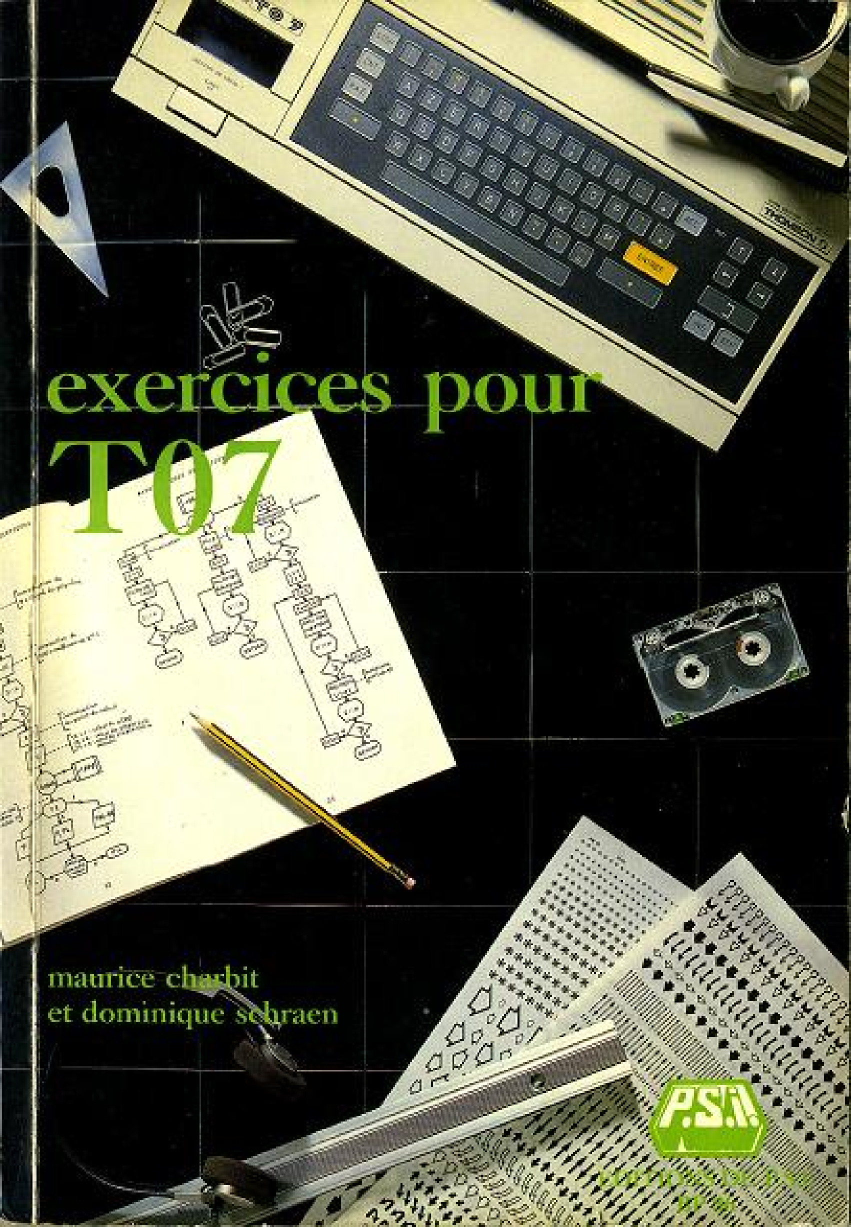

exercices pour
T07
maurice charbit
et dominique schraen
P.S.I.

Exercices pour T07

La collection « MATERIELS » s'intéresse à l'utilisation de tel ou tel type d'ordinateur, depuis la première prise de contact jusqu'à l'utilisation la plus pointue. Elle est aujourd'hui constituée de 34 titres

— Exercices sur Apple II — Frédéric Lévy
— La découverte de l'Apple II — Dominique Schraen et Frédéric Lévy
— La pratique de l'Apple II — volume I — Nicole Bréaud-Pouliquen
— La pratique de l'Apple II — volume II — Nicole Bréaud-Pouliquen
— La pratique de l'Apple II — volume III — Nicole Bréaud-Pouliquen et Daniel-Jean David
— La découverte de l'Atari — Daniel-Jean David
— La pratique du CBM — volume II — Daniel-Jean David
— La découverte du Commodore 64 — Daniel-Jean David
— L'assembleur du Commodore 64 — Daniel-Jean David
— La pratique du Commodore 64 — Daniel-Jean David
— Dragon tout feu tout flammes — Trévor Toms et John Phipps — traduit par Olivier Arnaud
— La découverte du FX-702 P — Jean-Pierre Richard
— Exercices pour Goupil — Yves Martin
— Programmer HP-41 — Philippe Deschamps et Jean-Jacques Dhénin
— La découverte du PB-100 — Pierrick Moigneau
— La découverte du PC-1251 — Jean-Pierre Richard
— La découverte du PC-1500 — Jean-Pierre Richard
— Le petit livre du Spectrum — Trévor Toms — traduit par Allan Keil
— Exercices pour ZX-Spectrum — Julien Lévy
— La pratique du Spectrum — tome 1 — Xavier Linant de Bellefonds
— La pratique du Spectrum — tome 2 — Marcel Henrot
— La découverte du T07 — Dominique Schraen et Maurice Charbit
— Exercices pour T07 — Maurice Charbit et Dominique Schraen
— Exercices sur TRS-80 — Frédéric Lévy
— Les graphiques sur TRS-80 — Donn Inman — traduit par Alain Pinaud
— La découverte de la TI 57 — Xavier de la Tullaye
— La découverte du TI-99/4A — Frédéric Lévy et Dominique Schraen
— Exercices pour TI-99/4A — Frédéric Lévy
— La découverte du VIC — Daniel-Jean David
— La découverte du VIC — volume 1 — Daniel-Jean David
— La pratique du ZX-81 — tome 1 — Xavier Linant de Bellefonds
— La pratique du ZX-81 — tome 2 — Marcel Henrot
— Le petit livre du ZX-81 — Trévor Toms — traduit par Ghislaine Lapeyre
— La découverte de l'Oric — Daniel-Jean David
— La découverte de l'Alice et MC/10 — Maurice Charbit

D'autres ouvrages relatifs au T07 :

— Basic pour tous — Sophie Brébion et Jacques Boisgontier
  Collection « ... Pour tous »
— Basic Microsoft — Ken Knecht
  Collection « LANGAGES »
— Jeux, trucs et comptes pour T07 — Michel Benelfoul
  Collection « PROGRAMMES »
— Le T07 à l'école — Daniel Nielsen
— Découverte du T07 — Dominique Schraen et Maurice Charbit
  Collection « MATERIELS »

RAPPELS

Les séries :

En fait, il faudrait parler de niveaux, puisque la couleur attachée à chaque ouvrage permet de situer la « force » de celui-ci selon le code suivant :

Série VERTE : ouvrage d'initiation ne nécessitant que des connaissances de base.
Série BLEUE : suppose une connaissance élémentaire du sujet traité.
Série ROUGE : ouvrage d'approfondissement, niveau de complexité moyen.
Série NOIRE : ouvrage d'approfondissement, niveau de complexité élevé.

Les collections :

Les ouvrages d'Edition du PSI, actuellement une centaine, sont répartis en collections :

« LANGAGES », **« MATERIELS »**, **« PROGRAMMES »**, **« GUIDES PRATIQUES »**, **« MEMENTOS »**, **« UTILISATIONS DE L'ORDINATEUR »**, **« LOGIGUIDE »**, **« METHODES PRATIQUES »** et pour l'initiation, outre quelques livres hors collection, « ... **POUR TOUS** ».

ISBN : 2-86595-147-2

Exercices pour TO7

par
Maurice Charbit
et
Dominique Schraen



Editions du P.S.I.
1984

PRÉSENTATION DES AUTEURS

Maurice CHARBIT *est ingénieur et enseignant à l'Ecole Nationale Supérieure des Télécommunications.*

Dominique SCHRAEN *est informaticien et l'auteur de :*

- La découverte de l'Applesoft - tome 1 (avec Frédéric LEVY)
- La découverte du TI-99/4A (avec Frédéric LEVY)
- La découverte du DRAGON (avec Frédéric LEVY)

Maurice CHARBIT ET Dominique SCHRAEN *ont déjà écrit en collaboration :*

- La découverte du TO 7

SOMMAIRE

INTRODUCTION

Ce livre s'adresse à tous ceux qui ont une connaissance élémentaire du Basic et sont désireux de la mettre en pratique sur des exemples concrets. Il permettra ainsi aux lecteurs de maîtriser le jeu d'instructions par la résolution d'exercices classiques de programmation.

Le livre comprend une première partie donnant les énoncés des exercices et une seconde partie réservée aux solutions.

La première partie est divisée en chapitres consacrés chacun à un thème donné.

Les énoncés sont en général suivis de l'analyse détaillée du problème et parfois même de la solution lorsque l'exercice traité constitue un exemple introduisant une notion nouvelle et importante.

Pour tirer le meilleur profit de ces exercices il est conseillé d'essayer de résoudre le problème uniquement à partir de l'énoncé sans se référer, dans un premier temps, à l'analyse proposée.

La solution donnée pour chaque exercice dans la seconde partie du livre sous forme de listing commenté n'est évidemment pas la seule possible, il est d'ailleurs recommandé au lecteur de la comparer de façon critique aux solutions qu'il aura trouvées.

PRÉSENTATION DES EXERCICES

*Les **énoncés** sont formés, en général, de trois rubriques :*

- **PROBLEME** : *donnant l'objectif du programme ;*
- **ENTREE** : *décrivant les données que l'utilisateur du programme devra fournir (par exemple un nom à taper au clavier) ;*
- **SORTIE** : *indiquant ce que le programme affichera sur l'écran.*

*L'**analyse** est constituée d'un organigramme commenté.*

*Les **solutions** comprennent le listing du programme proposé, la liste des variables qu'il utilise et l'explication détaillée des lignes importantes du programme.*
Le numéro de la ligne précède alors son explication.
(Lorsque l'explication concerne un groupe de ligne concécutives seuls les numéros de la première et de la dernière ligne sont indiqués, séparés par un tiret).

1re PARTIE

ÉNONCÉS ET ANALYSES

CHAPITRE I

AFFICHAGE, CALCULS SIMPLES, TIRAGE DE NOMBRES ALÉATOIRES

EXEMPLE 1.1

PROBLEME : affichage des initiales et de la longueur (nombre de lettres) du nom d'une personne.

ENTREE : le nom et le prénom de la personne écrits en majuscules.

SORTIE : les initiales et la longueur du nom.

Solution

```
10 CLS
20 INPUT "Quel est votre nom";N$
30 INPUT "Quel est votre Prenom";P$
40 PRINT:PRINT
50 PRINT TAB(10)LEFT$(N$,1);".";LEFT$(P$,1);"."
60 PRINT
70 PRINT "LONGUEUR DU NOM :";LEN(N$)
80 END
```

Variables

N$ - nom
P$ - prénom

Commentaires

40, 60 : passe des lignes pour faciliter la lecture.

50 : affichage à partir de la colonne 10 des initiales suivies d'un point.

EXERCICES POUR TO7

EXERCICE 1.2

ENTREE : le prénom et l'année de naissance d'une personne.

SORTIE : affichage du texte :

```
EH BIEN BONJOUR (prénom)
JE NE REPETERAI A PERSONNE QUE TU AS
AUJOURD'HUI (âge) ANS.
```

Analyse

Il faut :

1) demander le prénom de la personne
2) demander son année de naissance
3) afficher les lignes de texte (en calculant l'âge).

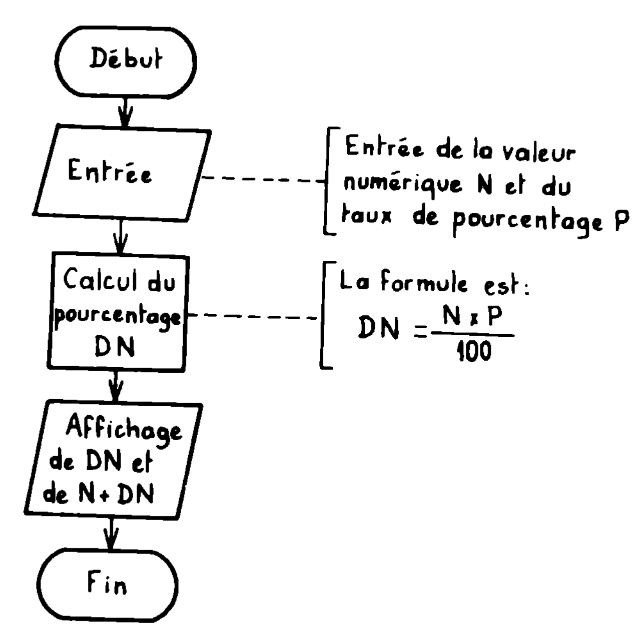
EXERCICE 1.3

PROBLEME : calcul du pourcentage P d'une valeur numérique N et de la valeur obtenue par application de ce pourcentage.

ENTREE ; la ~~#~~ valeur numérique et le taux de pourcentage.

SORTIE la ~~#~~ variation de la valeur numérique et la valeur finale.

Analyse



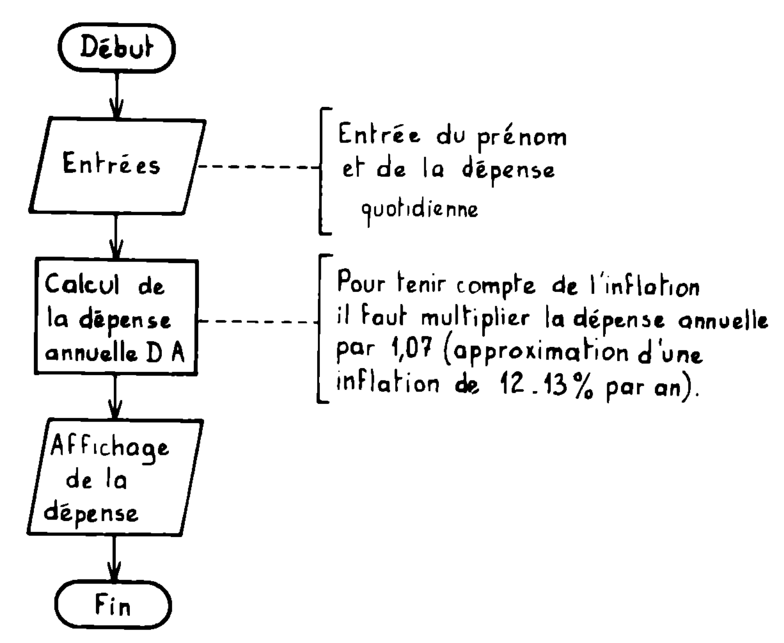
EXERCICE 1.4

PROBLEME : calcul de la dépense annuelle approximative pour votre nourriture, à partir de la dépense quotidienne moyenne (en tenant compte de l'inflation).

ENTREE la dépense quotidienne pour votre nourriture.

SORTIE : la dépense annuelle (essayez de rendre l'affichage aussi clair que possible grâce aux instructions CLS et LOCATE).

Analyse



EXERCICE 1.5

PROBLEME : le joueur doit découvrir par essais successifs un nombre entré auparavant par une autre personne.

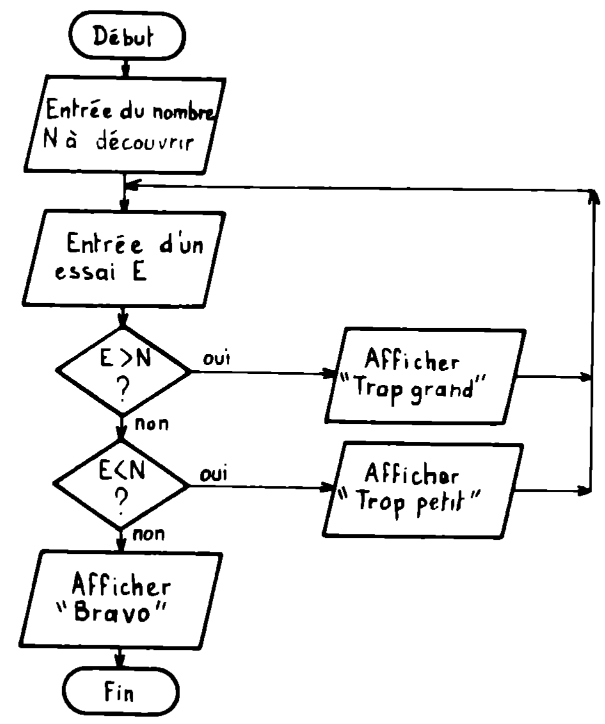
ENTREE : le nombre à découvrir (entier compris entre Ø et 1ØØØ) puis les essais.

SORTIE : un des messages, "TROP GRAND", "TROP PETIT", ou "BRAVO, VOUS AVEZ TROUVE" selon l'essai.

Analyse

Après l'entrée de l'essai, celui-ci est testé par

rapport au nombre à découvrir : s'il lui est supérieur ou inférieur le message correspondant est affiché et un nouvel essai est proposé. Sinon, le message "BRAVO, VOUS AVEZ TROUVE" s'affiche et le programme s'arrête.



EXEMPLE 1.6

PROBLEME : simulation du lancer d'un dé.

SORTIE : un nombre entier aléatoire compris entre 1 et 6.

Analyse

RND donne un nombre aléatoire compris entre Ø (inclus) et 1 (exclu). Pour obtenir un nombre entier aléatoire compris entre 1 et 6, il faut multiplier par 6 le nombre fourni par RND, puis prendre la partie entière du résultat et lui ajouter 1. L'instruction correspondante à ce calcul est : INT(6 * RND) + 1.

Solution

```
10 CLS
20 D=INT(6*RND)+1
30 ATTRB 1,1
40 PRINT
```

```
50 PRINT TAB(8)D
60 PRINT
70 ATTRB 0,0
80 INPUT "VOULEZ-VOUS LANCER LE DE A NOUVEAU (OUI/NON
) ";A$
90 PRINT
100 IF A$<>"NON" THEN 20
110 END
```

Variables

D - nombre entier aléatoire entre 1 et 6

A$ - réponse du joueur.

Commentaires

20 : calcul du nombre aléatoire

30 : provoque l'affichage en doubles hauteur et largeur des messages suivants.

70 : rétablit l'affichage habituel.

80 : question posée à l'utilisateur (les deux réponses possibles sont incluses dans le texte affiché).

100 : si l'utilisateur n'a pas répondu non, alors l'exécution du programme continue à la ligne 20.

PETIT EXERCICE 1.7

ENTREE : un nombre de dés à jeter (entre 1 et 8).

SORTIE : la somme des nombres sortis.

EXERCICE 1.8

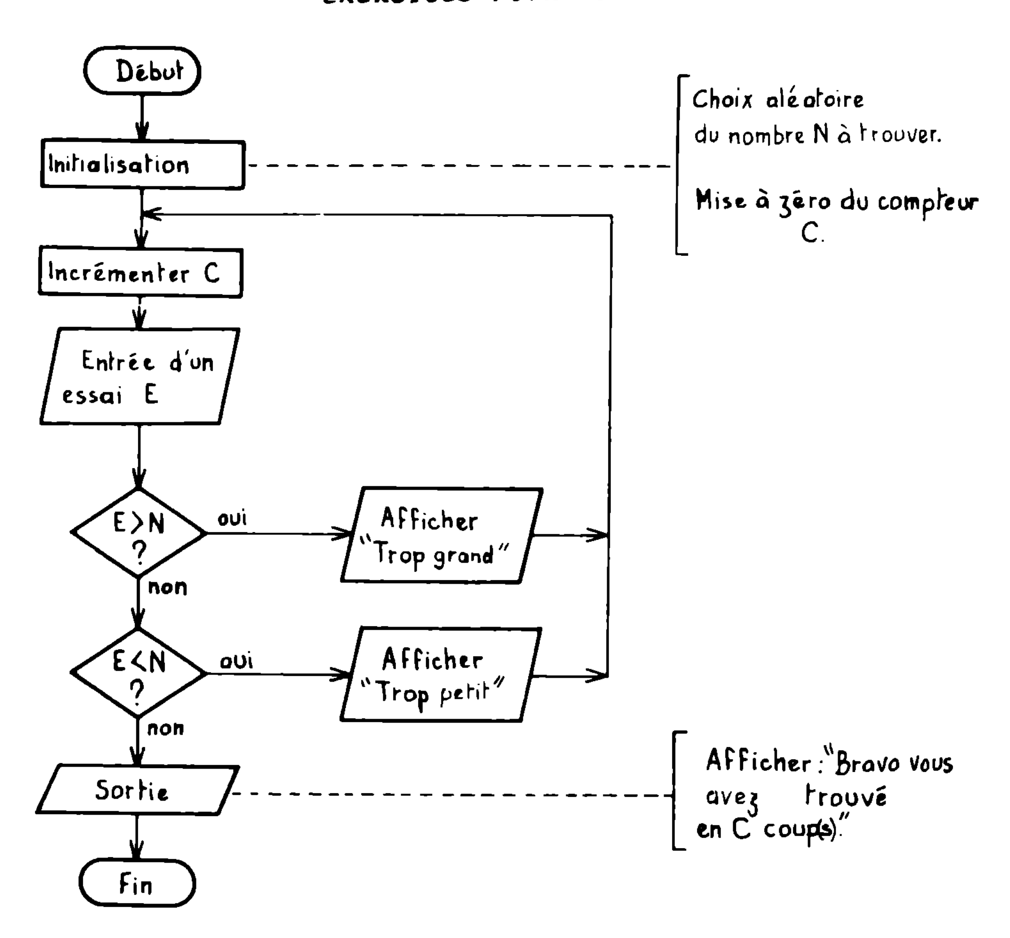
PROBLEME : version améliorée de l'exercice 1.5 Le programme choisit au hasard le nombre à découvrir et compte le nombre d'essais nécessaires au joueur pour arriver à la solution.

ENTREE : les essais.

SORTIE : selon l'essai, un des messages : "TROP GRAND", "TROP PETIT" ou "BRAVO VOUS AVEZ TROUVE EN C COUP(S)".

Analyse

Le nombre d'essais est contenu dans la variable C. Pour cela, elle est mise à zéro au début du programme puis, à chaque essai, elle est augmentée de 1 (elle est incrémentée). Une telle variable est appelée "compteur".



EXERCICE 1.9

PROBLEME : version améliorée de l'exercice 1.8 ajoutant une initialisation aléatoire de l'instruction RND.

ENTREES une touche du clavier puis les essais.

SORTIE selon l'essai, un des messages : "TROP GRAND", "TROP PETIT" ou "BRAVO VOUS AVEZ TROUVE EN C COUP(S)".!".

Analyse

L'initialisation aléatoire de la fonction RND utilise indirectement la fonction INKEY$.

CHAPITRE II

MANIPULATION DE TABLEAUX (RECHERCHE, CALCULS, CLASSEMENT, INSERTION)

EXEMPLE 2.1

PROBLEME : trouver le nombre de valeurs positives ou nulles dans une liste de dix valeurs.

ENTREE : les dix valeurs.

SORTIE : le nombre de valeurs positives ou nulles.

Analyse

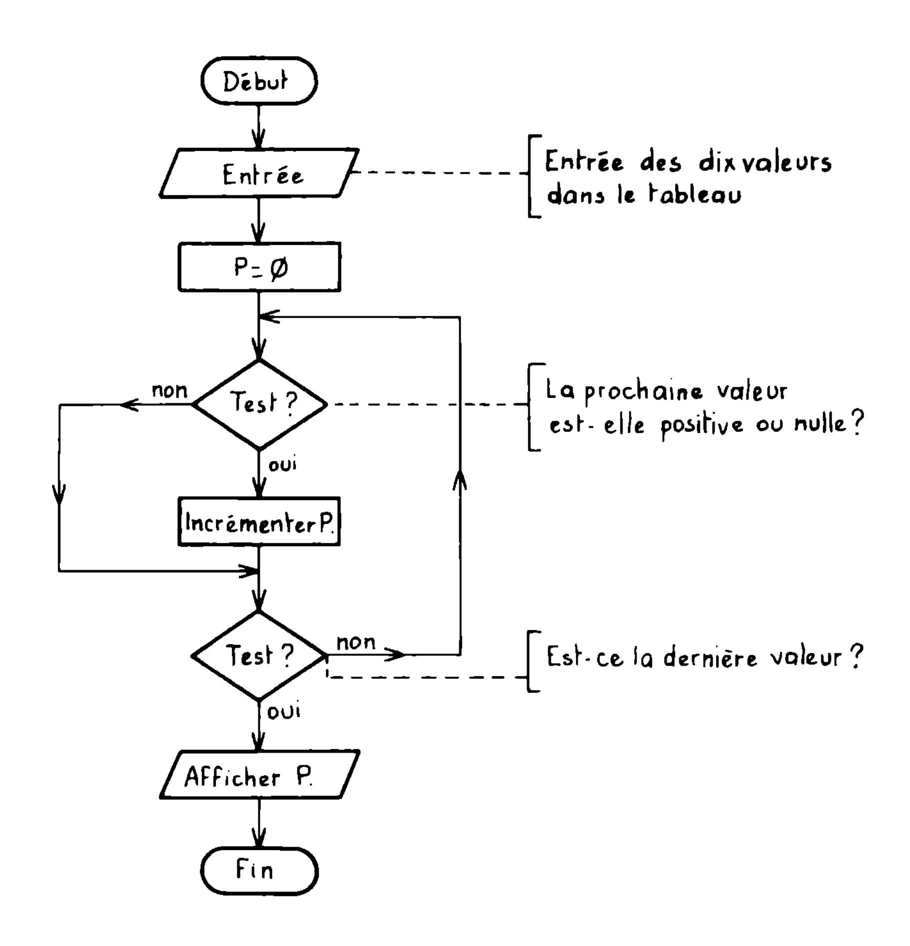
Il est nécessaire d'utiliser un compteur (ici P) qui sera incrémenté (augmenté de 1) dans la boucle pour chaque valeur positive trouvée.

L'utilisation d'un tableau, pour enregistrer les valeurs, facilite le traitement.

Solution

```
10 DIM V(10)
20 REM
30 REM CHOIX DES VALEURS
40 CLS
50 FOR I=1 TO 10
60 PRINT "VALEUR ";I,
70 INPUT V(I)
80 NEXT I
90 REM
100 REM POSITIFS OU NULS
110 P=0
120 FOR I=1 TO 10
130 IF V(I)>=0 THEN P=P+1
140 NEXT I
150 REM
160 REM AFFICHAGE
```

```
170 FOR I=1 TO 10
180 PRINT V(I)
190 NEXT I
200 PRINT
210 PRINT "NOMBRE DE VALEURS POSITIVES OU NULLES
P
220 END
```



Variables

I - compteur de boucle
P - nombre de valeurs positives

V(1Ø) - tableau des 1Ø valeurs.

Commentaires

1Ø : dimensionnement du tableau

5Ø-8Ø : boucle d'entrée des valeurs : le numéro de chaque valeur est affiché par l'instruction PRINT et la ligne 6Ø car il est impossible de faire afficher par l'instruction INPUT un message comprenant la valeur de I.

11Ø : initialisation du compteur de valeurs positives.

13Ø : si la Ième valeur est positive ou nulle, P est incrémenté.

17Ø-19Ø: les dix valeurs sont affichées avant le résultat.

EXERCICE 2.2

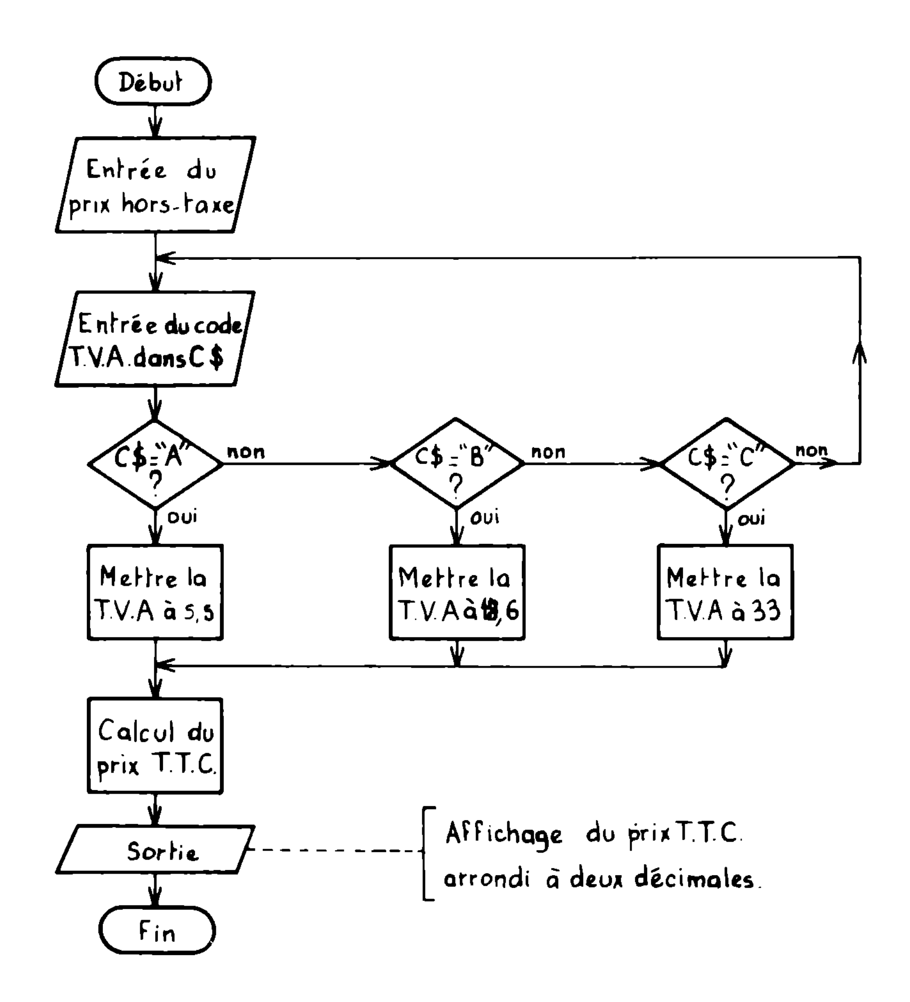
PROBLEME : calcul d'un prix TTC à partir du prix hors-taxe et du code de la TVA à lui appliquer.

ENTREE : Prix hors-taxe et code du taux de TVA (A, B ou C pour 5,5 %, 18,60 % ou 33 % respectivement.)

SORTIE : Le prix TTC arrondi à deux décimales.

Analyse

Le taux de TVA correspondant au code est affecté à une variable, puis le prix TTC est calculé à partir de cette variable. Si le code tapé ne correspond à rien, un autre code TVA est demandé.



EXERCICE 2.3

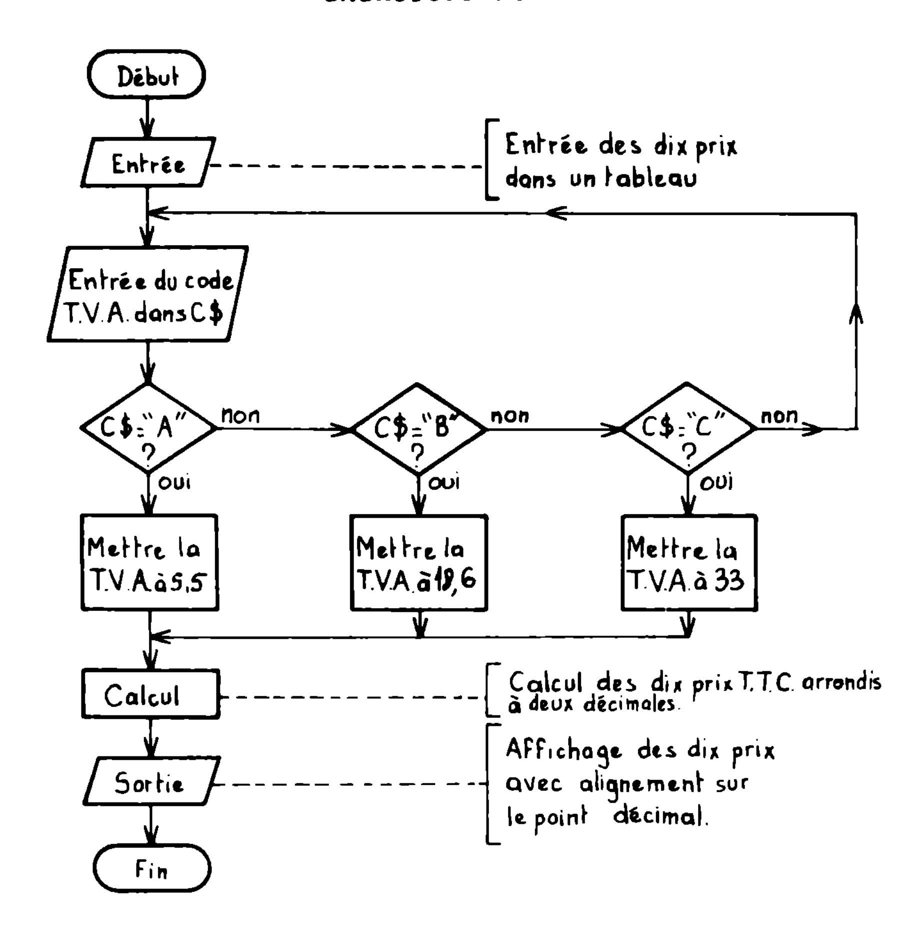
PROBLEME : Même programme que le précédent (exercice 2.2) mais pour un groupe de dix prix à la fois.

ENTREE : dix prix hors-taxe et le code du taux de TVA à leur appliquer.

SORTIE : les dix prix TTC arrondis à deux décimales et alignés sur le point décimal.

Analyse

L'utilisation d'un tableau pour enregistrer les dix prix permet de faciliter les entrées, l'affichage et les calculs, en utilisant des boucles FOR-NEXT.



PETIT EXERCICE 2.4

PROBLEME : Lecture dans un tableau d'une liste de mots enregistrée dans le programme sous forme de DATA (essayer de trouver plusieurs manières différentes de résoudre le problème).

ENTREE : La liste des mots (à partir de lignes DATA).

SORTIE : Affichage du tableau.

EXERCICE 2.5

PROBLEME : A partir d'une liste enregistrée de noms de personnes avec leur prénom, chercher la présence d'une personne dans cette liste.

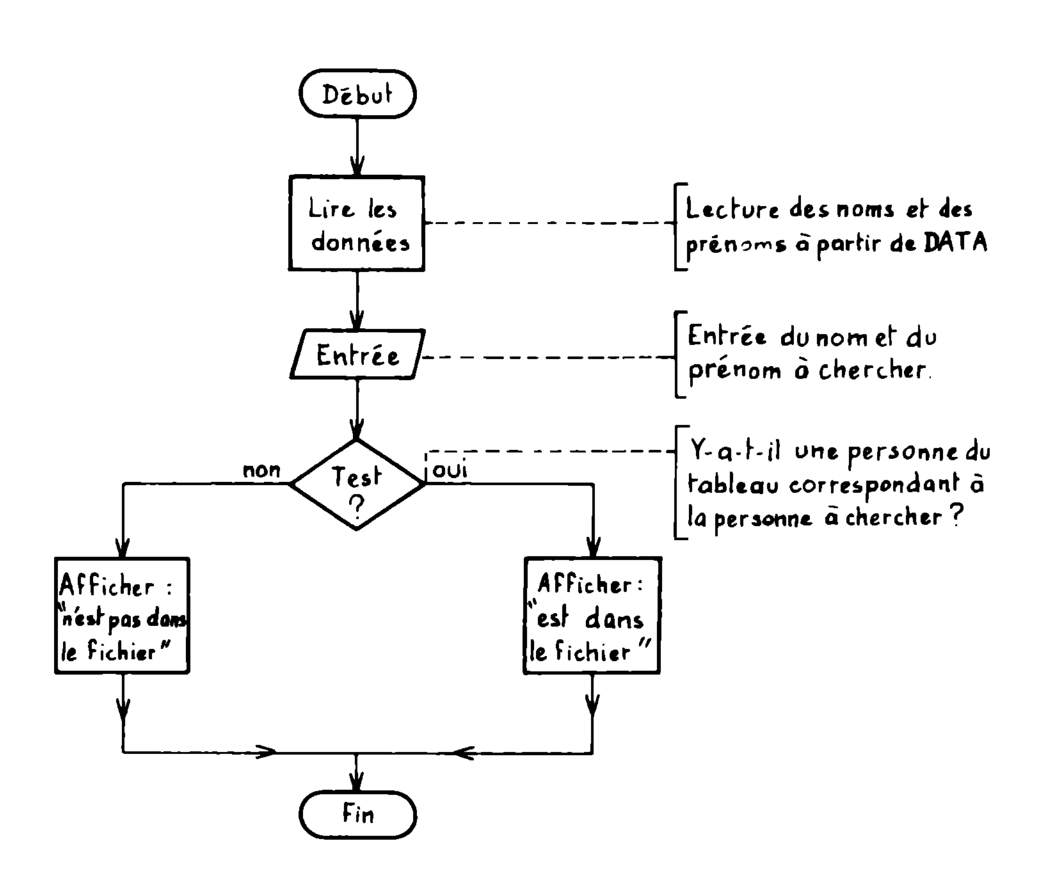
ENTREE : Une liste de personnes (noms et prénoms). Une personne à chercher (nom et prénom).

SORTIE : Un message indiquant si la personne se trouve, ou ne se trouve pas, dans la liste.

Analyse

L'utilisation de l'instruction DATA permet une lecture facile et rapide des données dans un tableau.

Pour savoir si la personne se trouve dans la liste, il faut comparer successivement son nom et son prénom à ceux de toutes les personnes.



EXERCICE 2.6

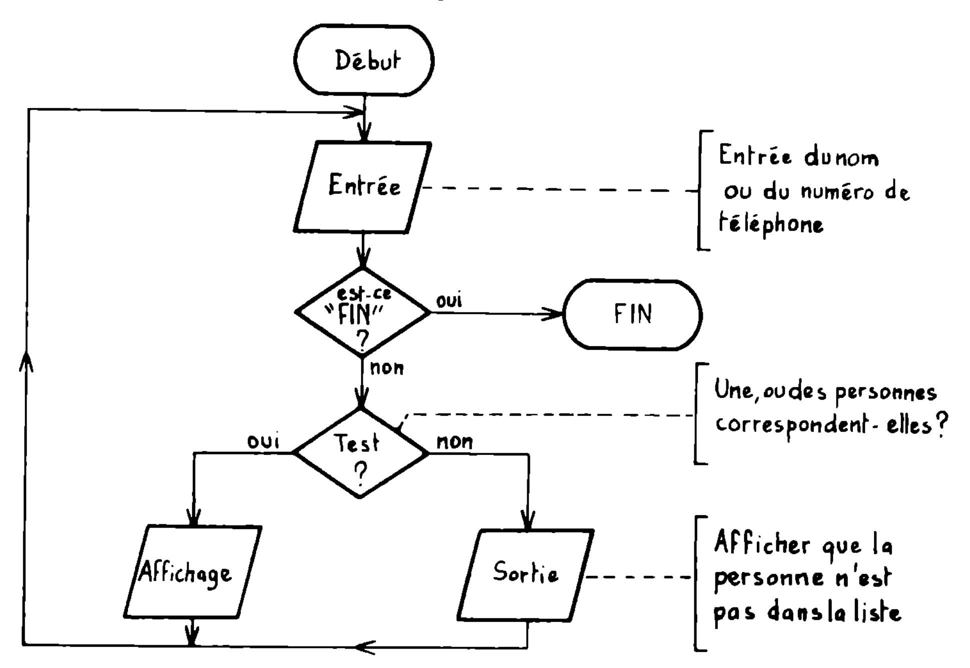
PROBLEME : Recherche, dans un agenda, d'un numéro de téléphone ou d'un nom.

ENTREE : Le nom de la personne dont on souhaite connaître le numéro de téléphone, ou le numéro de celle dont on cherche le nom.

SORTIE : La personne recherchée avec son numéro de téléphone, si elle se trouve effectivement dans l'agenda (attention, plusieurs personnes peuvent porter le même nom !).

Analyse

Les personnes sont enregistrées sous forme de DATA, avec leur numéro de téléphone.



EXERCICE 2.7

PROBLEME : Trouver le plus petit et le plus grand élément d'une liste de nombres.

ENTREE : Une liste de nombres.

SORTIE : Le plus grand et le plus petit élément de la liste.

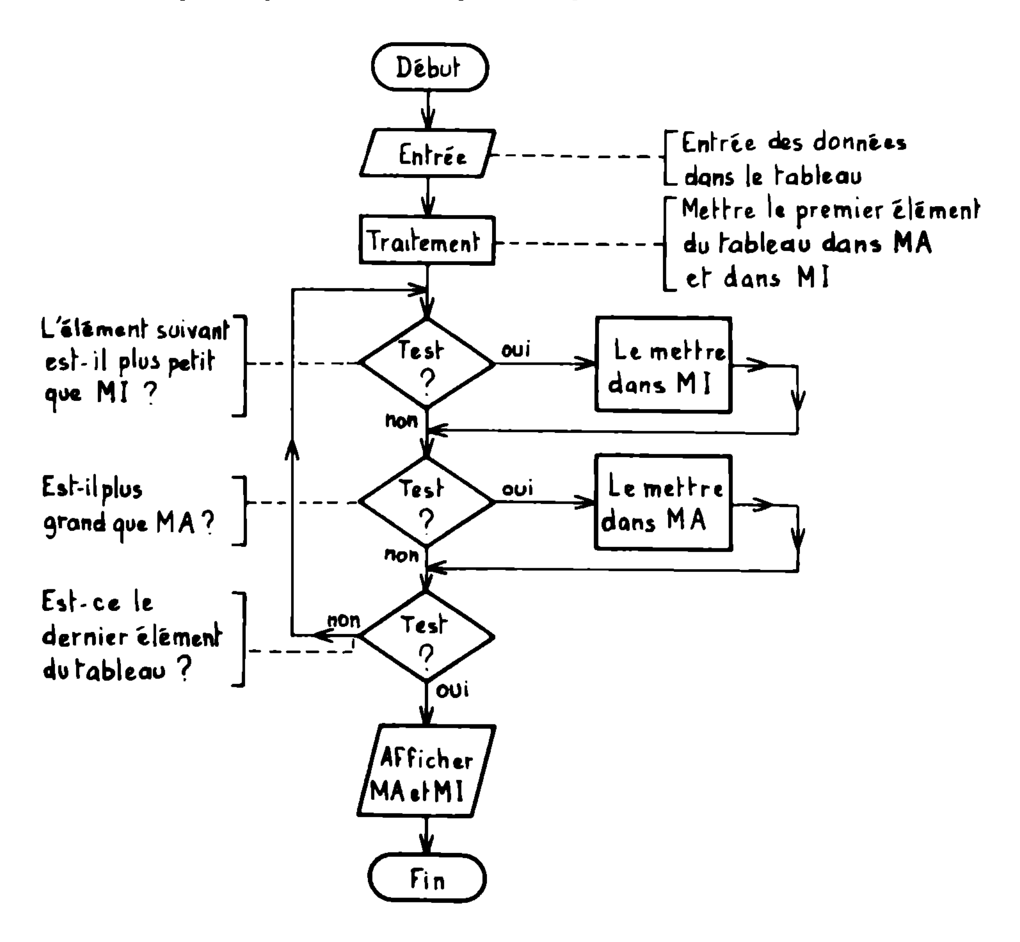
Analyse

Le plus grand élément du tableau est trouvé de la façon suivante :

1) mettre le premier élément du tableau dans la variable numérique MA ;
2) comparer un à un les éléments du tableau avec MA. Si l'élément est plus grand que MA alors c'est lui qu'il faut garder dans MA, sinon il ne faut pas changer la valeur de MA.

Lorsque tous les éléments du tableau ont été comparés, MA contient la valeur la plus grande.

Pour trouver l'élément le plus petit, le processus est le même (avec la variable MI), en gardant dans MI l'élément le plus petit à chaque comparaison.



EXERCICE 2.8

PROBLEME : Trouver dans une liste de personnes la première par ordre alphabétique et indiquer son rang.

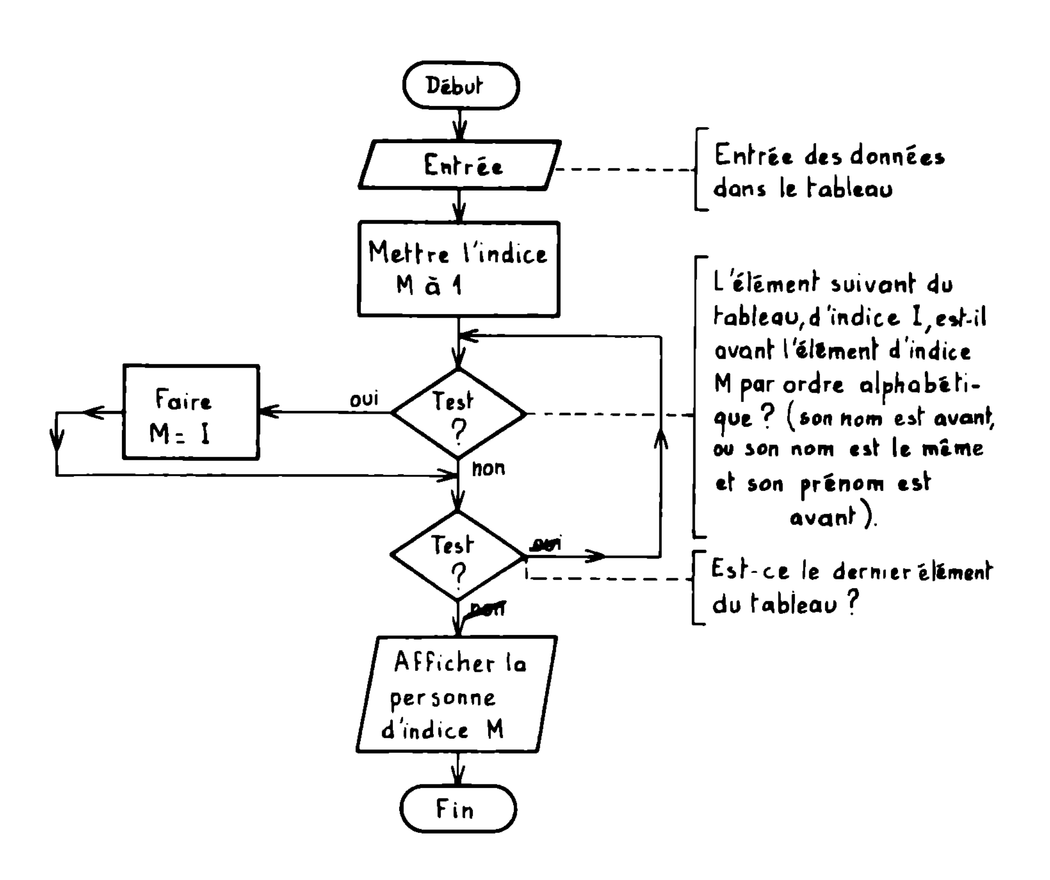
ENTREE : Une liste de personnes (noms et prénoms)

SORTIE : Le rang de la première personne, son nom et son prénom.

Analyse

La méthode pour trouver la première personne par ordre alphabétique est identique à celle donnée dans la solution de l'exercice précédent. Il faut en plus conserver le rang de l'élément cherché.

Il y a deux comparaisons à faire, l'une entre les noms, et l'autre, si les noms sont identiques, entre les prénoms.



EXERCICE 2.9

PROBLEME : Classement par ordre décroissant d'une liste de nombres entrés dans un ordre quelconque (utiliser deux tableaux, le premier contenant les nombres à classer, le second pour ranger les nombres classés).

ENTREE : Une liste de nombres.

SORTIE : La liste classee par ordre décroissant.

Analyse

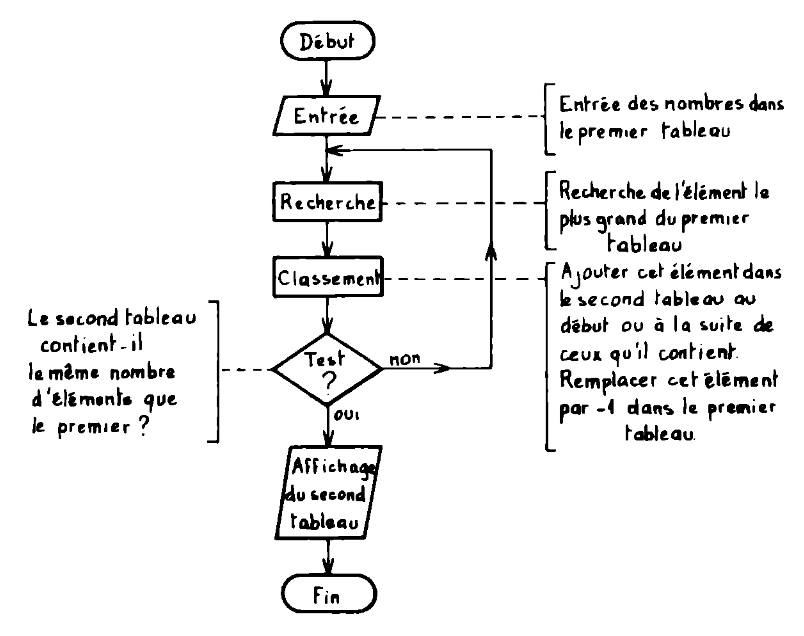
Deux manières de résoudre le problème sont proposées :

1ère méthode :

Elle nécessite l'utilisation d'un nombre particulier ne pouvant appartenir à la liste. Ici nous supposerons que tous les nombres à classer sont positifs ou nuls ; un nombre négatif, par exemple -1, ne peut donc lui appartenir.

Le programme recherche l'élément le plus grand du premier tableau qu'il ajoute au second tableau ; cet élément est ensuite remplacé par -1 dans le premier tableau; le processus est recommencé autant de fois qu'il y a de nombres à classer.

A la fin, le premier tableau ne contient plus que des -1, et le second tableau contient les nombres classés par ordre décroissant.

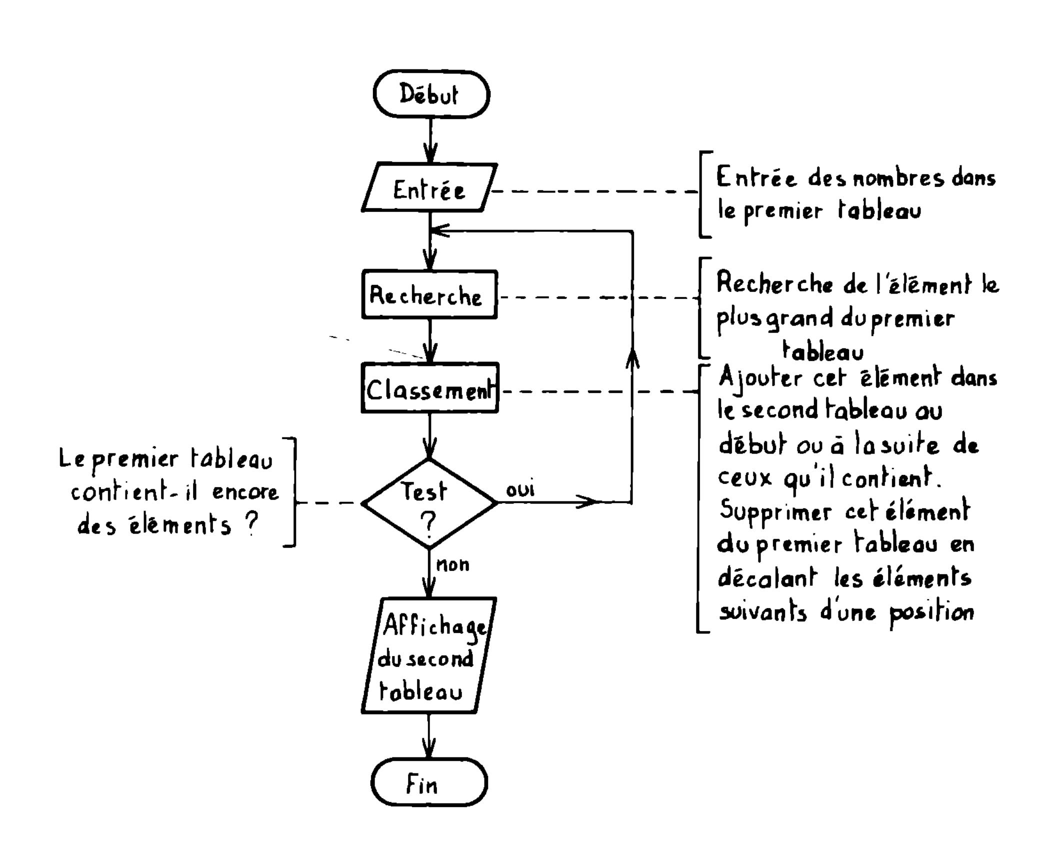


2ème méthode

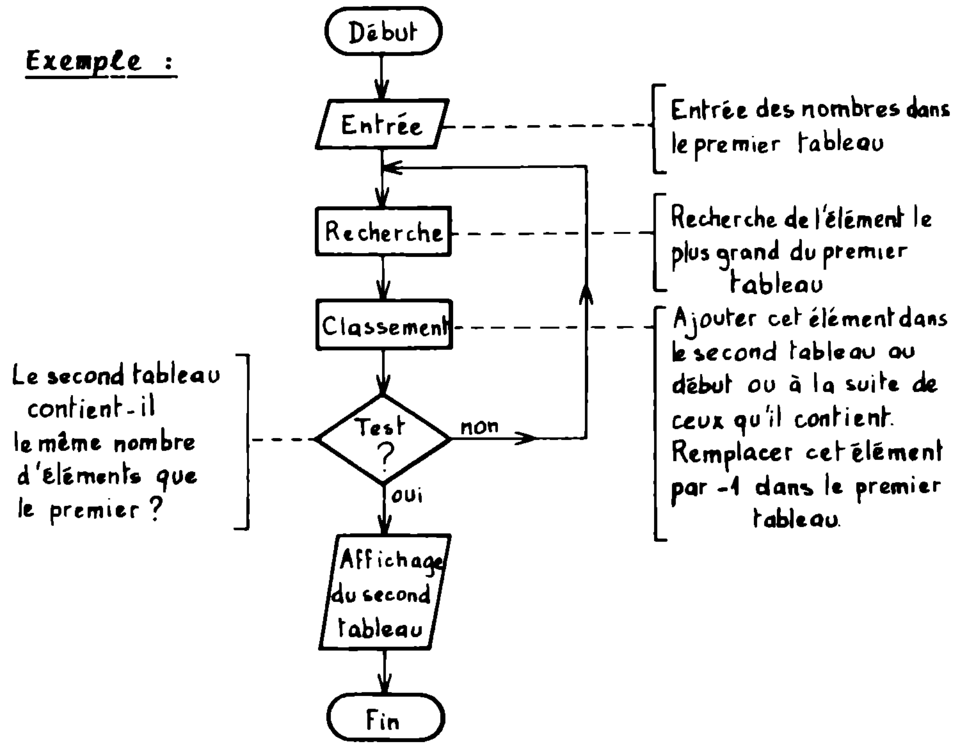
Cette méthode est applicable quels que soient les nombres à classer.

Après avoir trouve l'élément le plus grand du premier tableau et l'avoir transféré dans le second, celui-ci est supprimé du premier tableau en faisant "remonter" les éléments suivants d'une position; le tableau des nombres à classer contient donc un élément de moins.

Le processus est recommencé jusqu'à ce qu'il n'y ait plus d'éléments dans le premier tableau.



Exemple :



La première méthode a l'avantage d'être simple, mais elle nécessite l'existence d'une valeur ne pouvant appartenir au tableau initial (ici -1).

La deuxième méthode peut s'appliquer à un tableau quelconque mais est plus compliquée et plus longue à exécuter, à cause des décalages à effectuer.

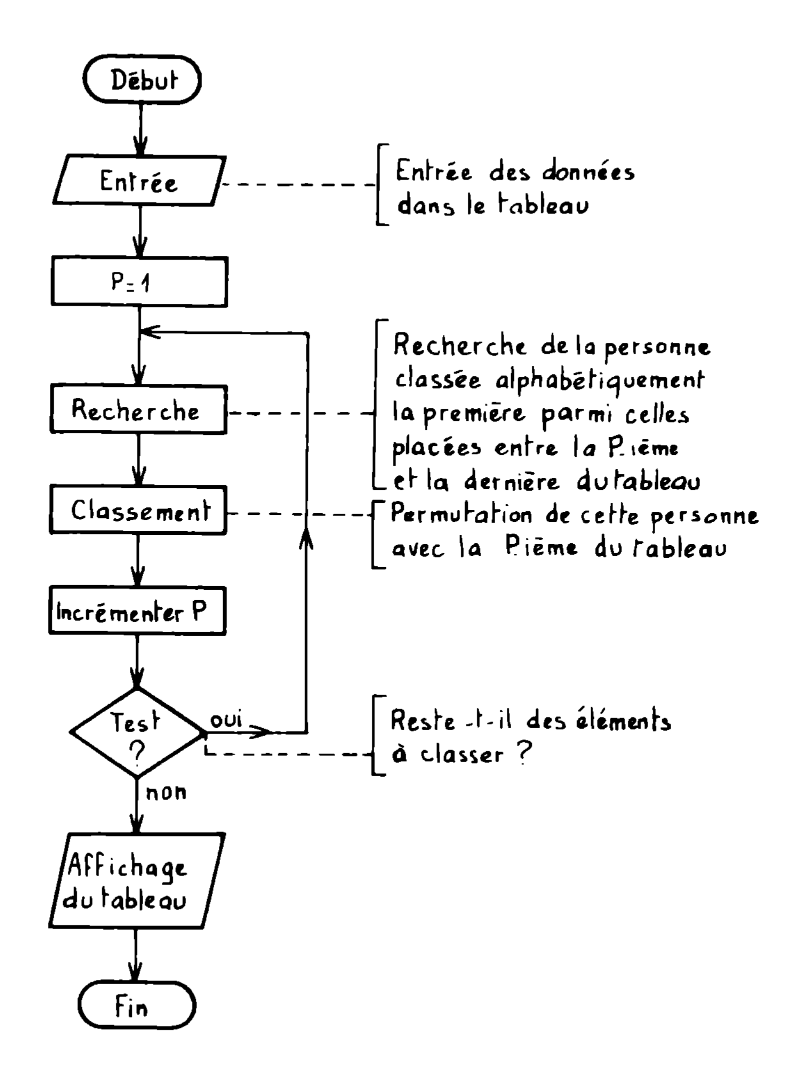
EXERCICE 2.10

PROBLEME : Même programme que le précédent (exercice 2.9) mais pour une liste de personnes à classer par ordre alphabétique, et en n'utilisant qu'un tableau.

ENTREE : Une liste de personnes (noms et prénoms).

SORTIE : La liste classée par ordre alphabétique

Analyse



Il est parfois nécessaire de n'utiliser qu'un tableau, par exemple lorsque la capacité de la mémoire du micro-ordinateur est insuffisante pour en contenir plusieurs.

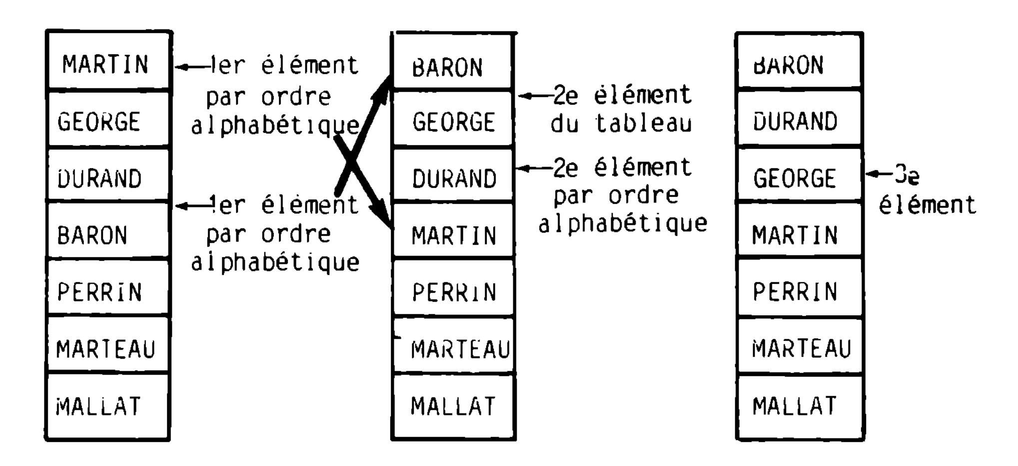
Voici la méthode utilisée :

Rechercher le premier élément du tableau par ordre alphabétique, puis permuter celui-ci avec le premier élément du tableau.

Le premier élément du tableau est alors aussi le premier par ordre alphabétique.

Recommencer avec les éléments suivants, chercher le premier par ordre alphabétique parmi eux, puis permuter celui-ci avec celui d'indice le plus petit.

Exemple :



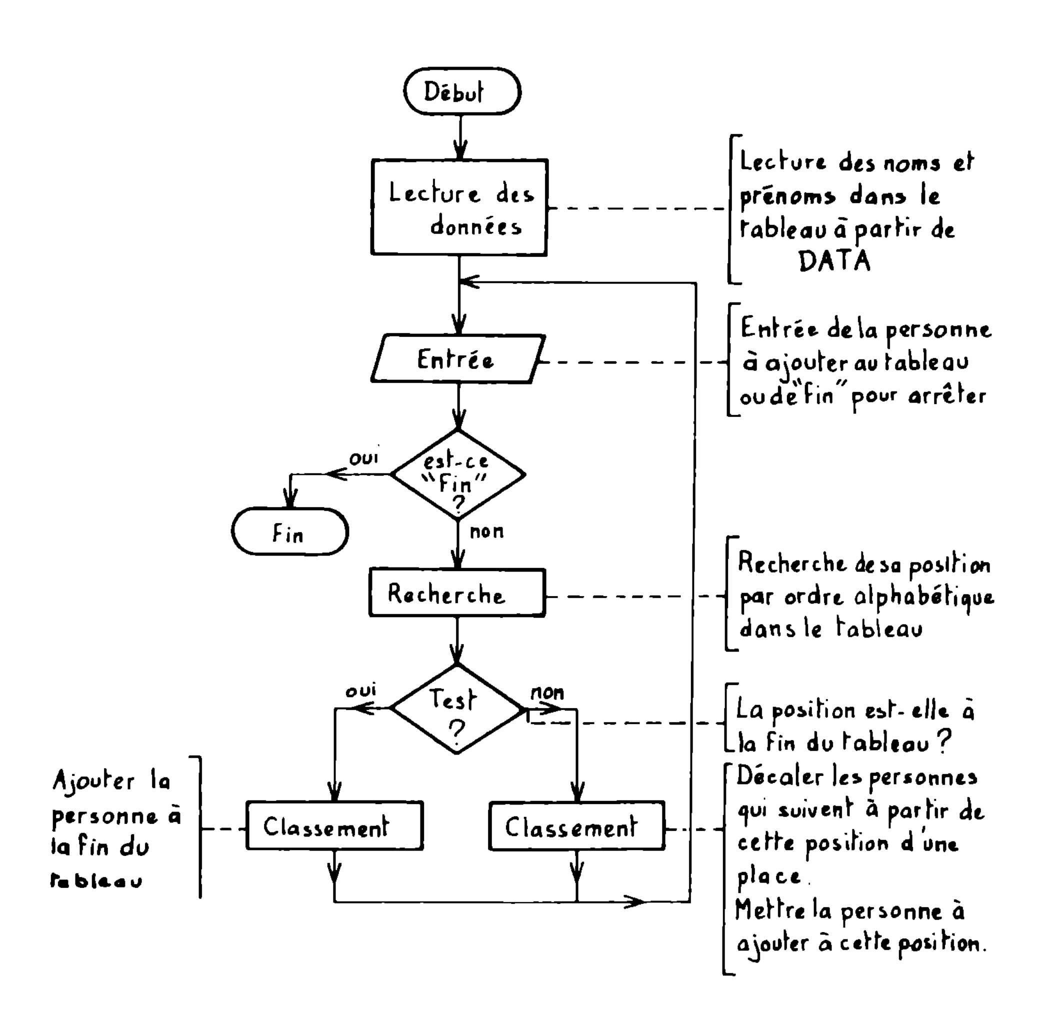
EXERCICE 2.11

PROBLEME : Ajouter une personne à un tableau classé alphabétiquement, en l'insérant à sa position dans l'ordre alphabétique.

ENTREE : Une liste de personnes (noms et prénoms) classée par ordre alphabétique. Une personne à insérer.

SORTIE : Affichage du nouveau tableau contenant la personne insérée à sa position dans l'ordre alphabétique.

Analyse



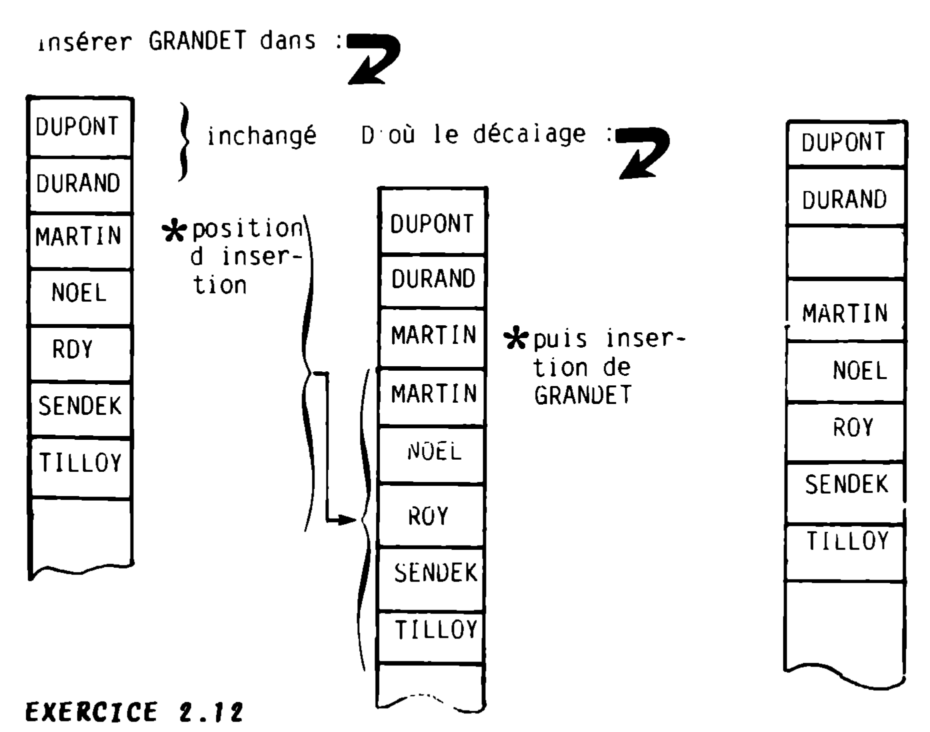
La liste des personnes enregistrée dans les instructions DATA est écrite préalablement dans l'ordre alphabétique.

Pour insérer la personne à sa bonne place dans l'ordre alphabétique, il faut d'abord trouver la position qu'elle occupera dans le tableau. Si cette position est la dernière du tableau, il suffit d'ajouter la personne à la fin, sinon il faut décaler d'une position toutes les

personnes qui la suivent dans l'ordre alphabétique.

Exemple :

insérer GRANDET dans :



EXERCICE 2.12

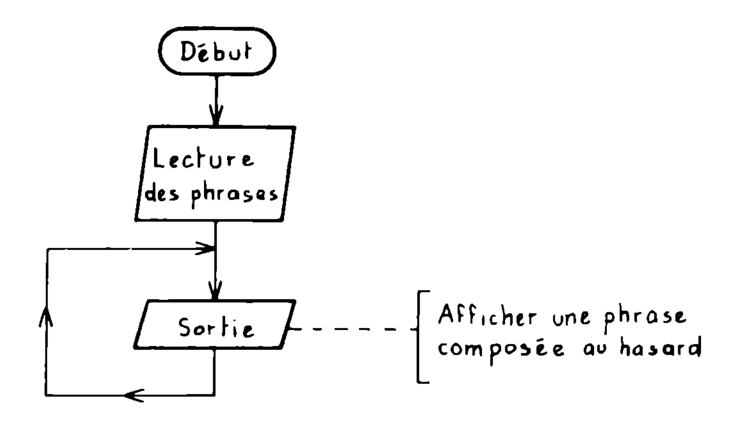
PROBLEME : Créer des phrases, composées aléatoirement à partir de noms, verbes et compléments.

ENTREE : Une liste de noms, verbes et compléments (en DATA).

SORTIE : Des phrases aléatoires.

EXERCICES POUR TO7

Analyse



CHAPITRE III

TRAITEMENT DE CHAÎNES DE CARACTÈRES (RECHERCHE, INSERTION, SUPPRESSION DE CARACTÈRES)

PETIT EXERCICE 3.1

PROBLEME : Suppression de tous les caractères "espace" pouvant exister dans une chaîne de caractères.

ENTREE : Une ligne de texte.

SORTIE : La ligne sans aucun espace.

PETIT EXERCICE 3.2

PROBLEME : Ajouter un espace entre chaque lettre d'une chaîne de caractères.

ENTREE : Une chaîne de caracteres.

SORTIE : La chaîne de caractères avec un espace ajouté entre chaque caractère.

PETIT EXERCICE 3.3

ENTREE : Une phrase ou un mot.

SORTIE : Affichages successifs du texte tronqué d'un caractère supplémentaire à chaque fois.

Exemple : BONJOUR donne BONJOUR

```
BONJOUR
ONJOUR
NJOUR
JOUR
OUR
UR
R
```

EXERCICES POUR TO7

EXERCICE 3.4

ENTREE : Une phrase ou un mot.

SORTIE : Affichages successifs du texte. A chaque affichage, le premier et le dernier caractères du texte sont enlevés.

Exemple : BONJOUR donne BONJOUR
ONJOU
NJO
J

Analyse

Pour obtenir la présentation donnée en exemple, plusieurs méthodes sont possibles :

On peut remplacer les caractères supprimés en début de ligne par autant d'espaces, ou bien utiliser à l'affichage la fonction TAB (c'est cette dernière méthode qui sera utilisée dans la solution proposée).

EXERCICE 3.5

ENTREE : Une phrase

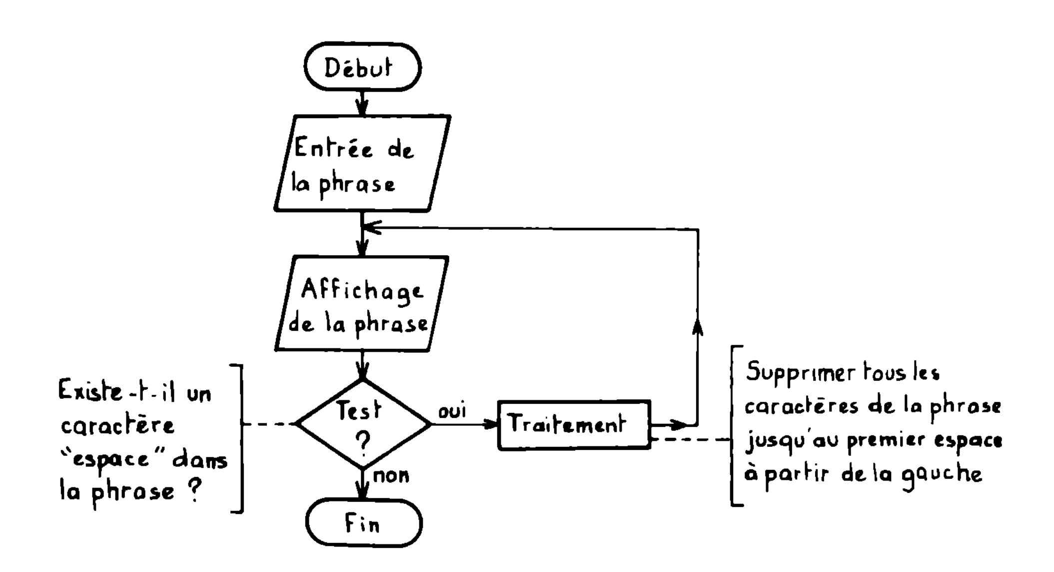
SORTIE : Affichages successsifs de la phrase tronquée d'un mot supplémentaire à chaque fois.

Exemple : LE CHAT EST SUR LE TOIT donne

LE CHAT EST SUR LE TOIT
CHAT EST SUR LE TOIT
EST SUR LE TOIT
SUR LE TOIT
LE TOIT
TOIT

Analyse

Nous supposerons que tous les mots sont séparés par un caractère "espace". Le programme va donc rechercher le premier espace à partir de la gauche de la phrase, supprimer tous les caractères jusqu'à cet espace, puis afficher la phrase, et recommencer jusqu'à ce qu'il ne reste plus de mots.



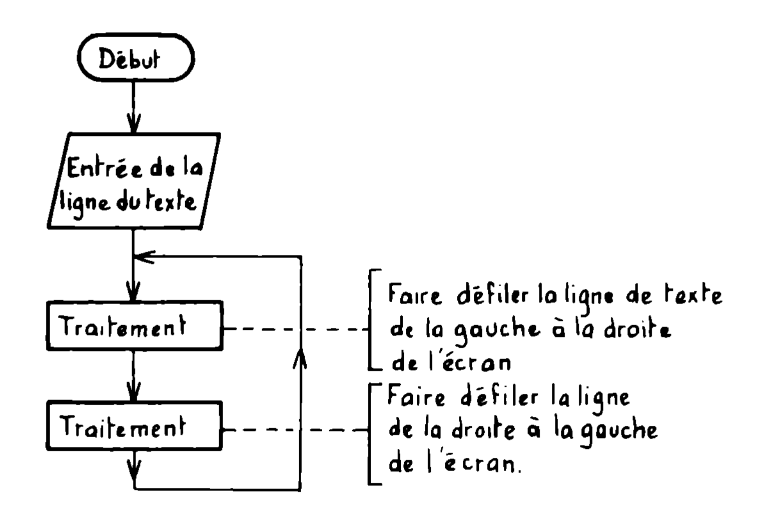
EXEMPLE 3.6

PROBLEME : Faire défiler une phrase sur l'écran de gauche à droite puis, lorsqu'elle atteint le bord, de droite à gauche, et ainsi de suite.

ENTREE : Une phrase de moins de 2Ø caractères.

SORTIE : Défilement de la phrase sur l'écran.

Analyse



Solution

```
10 CLS
20 LINE INPUT "LIGNE (MAX:20 CAR.) : ",L$
30 L=LEN(L$):IF L>20 THEN 20
40 REM DE GAUCHE A DROITE
50 FOR I=0 TO 40-L
60 PRINT TAB(I)L$;
70 FOR J=1 TO 10:NEXT J
80 NEXT I
90 REM DE DROITE A GAUCHE
100 FOR I=40-L TO 0 STEP -1
110 PRINT TAB(I)L$;
120 FOR J=0 TO 10:NEXT J
130 NEXT I
140 GOTO 40
```

Variables

I, J - compteurs

L - longueur de L$

L$ - ligne de texte

Commentaires

5Ø- 8Ø : La ligne est affichée à partir de la Ième position, successivement de la colonne Ø à la colonne 4Ø-L.

70 : Boucle de temporisation.

100-130 : La ligne est affichée à partir de la Ième position, successivement de la colonne 40-L à la colonne 0.

120 : Boucle de temporisation.

Remarque : Pour que le dernier caractère de L$ soit sur la dernière colonne de l'écran, il faut afficher L$ à partir de la colonne 40-L.

EXERCICE 3.7

PROBLEME : Faire défiler un texte sans arrêt de droite à gauche sur une même ligne d'écran.

ENTREE : Un texte de moins de 39 caractères.

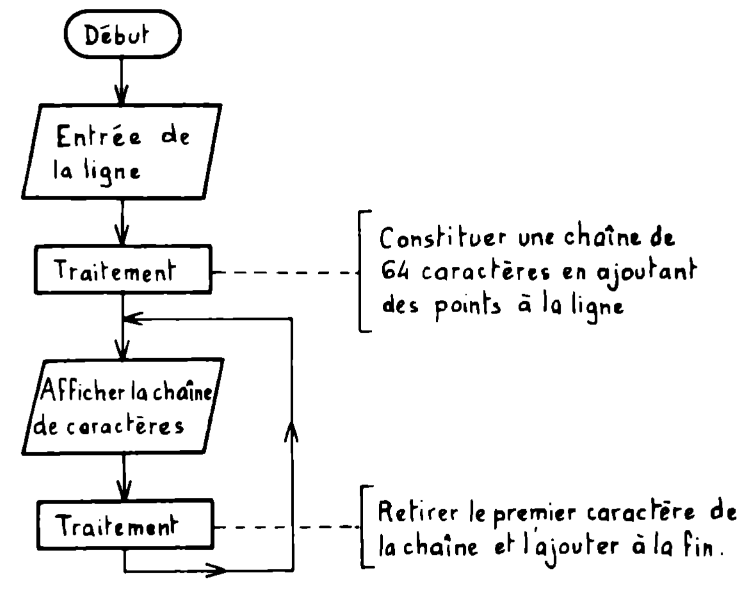
SORTIE : Faire défiler le texte sur l'écran.

Analyse

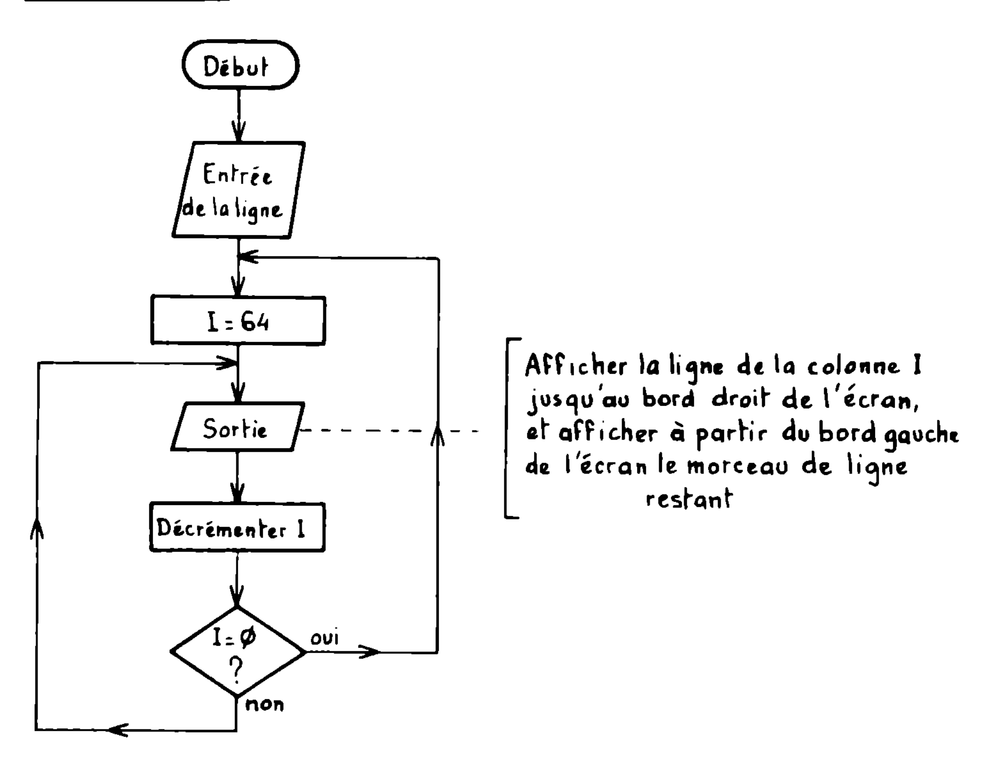
Deux méthodes sont proposées pour résoudre le problème

lère méthode :

Créer une chaîne de 40 caractères (largeur de l'écran) à partir du texte entré en lui ajoutant suffisamment de points à la fin. Pour obtenir le défilement, afficher la ligne, retirer le premier caractère de la chaîne, l'ajouter à la fin et recommencer.



2ème méthode :



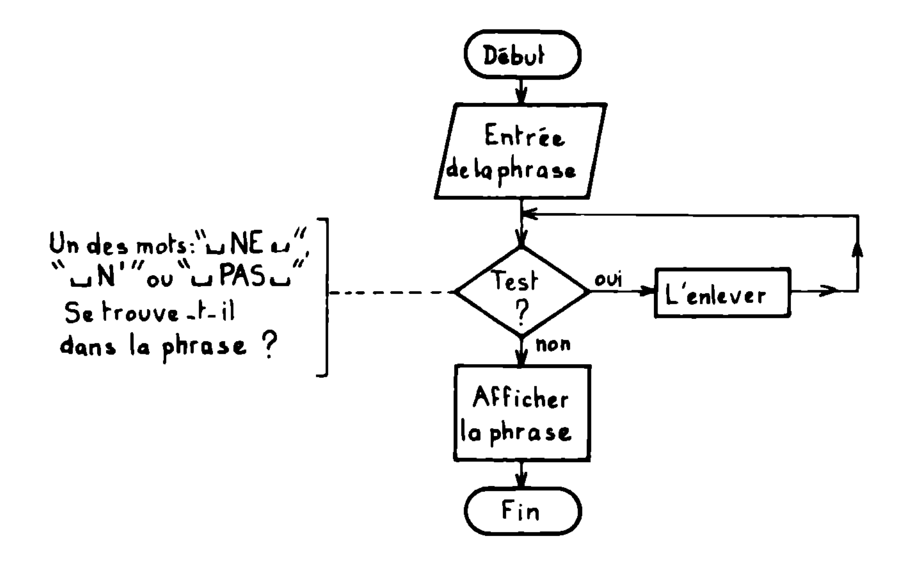
EXERCICE 3.8

PROBLEME : Transformer une phrase de la forme négative à la forme affirmative en supprimant de la phrase les mots NE, N' et PAS.

ENTREE : Une phrase à la forme négative (non terminée par un point)

SORTIE : La phrase à la forme affirmative.

Analyse



Remarque : Les comparaisons s'effectuent avec les mots NE, N' et PAS entourés d'espaces pour les distinguer de parties de mots (exemple : PASserelle). Attention au cas où la phrase commence avec un de ces mots, ou bien se termine avec le mot PAS.

PETIT EXERCICE 3.9

PROBLEME : Insertion de caractères dans une chaîne de caractères à une position donnée.

ENTREE : Une chaîne de caractères, puis la chaîne de caractères à insérer et la position d'insertion.

SORTIE : La nouvelle chaîne de caractères.

EXERCICE 3.10

PROBLEME : Remplacement d'un mot par un autre dans un texte ne dépassant pas 128 caractères. Le mot doit être remplacé autant de fois qu'il se trouve dans le texte (le texte modifié ne doit cependant pas dépasser 128 caractères).

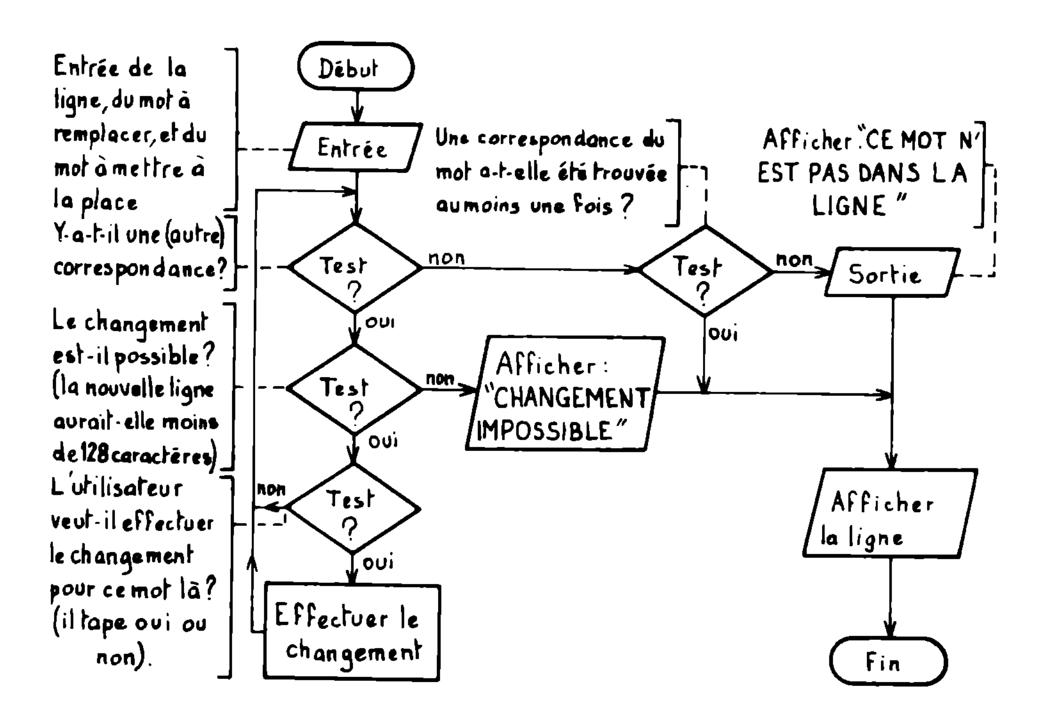
ENTREE : Un texte d'au plus 128 caractères. Un mot à remplacer et le mot à lui substituer.

SORTIE : Le nouveau texte avec les mots remplacés (il ne doit pas dépasser 128 caractères).

Analyse

La recherche dans le texte de la position du mot à remplacer est faite en le comparant à des morceaux de texte de même longueur, en commençant au début de la ligne, puis en se déplaçant d'un caractère jusqu'à la fin du texte.

A chaque fois que le mot à remplacer est trouvé, le programme demande à l'utilisateur, après le test de dépassement des 128 caractères, s'il veut ou non remplacer ce mot là dans le texte.



EXERCICE 3.11

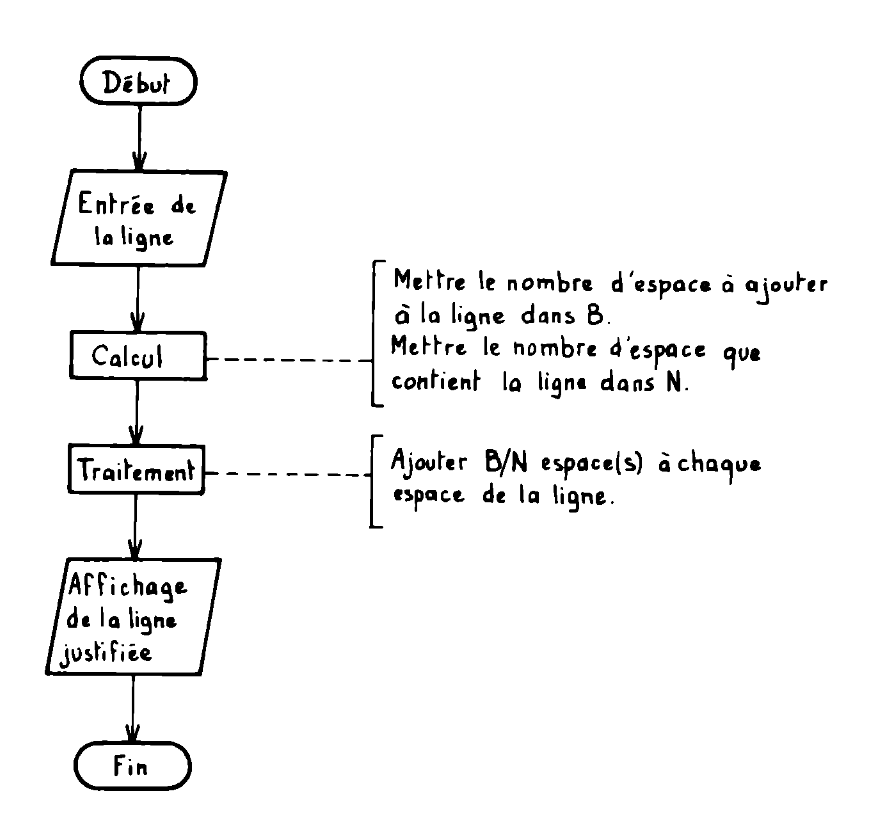
PROBLEME : Justification d'une ligne de texte sur l'écran.

ENTREE : Une ligne de texte de longueur inférieure ou égale à 4Ø caractères.

SORTIE : La ligne justifiée sur l'écran (des espaces ont été ajoutés entre chaque mot, de manière à ce que le premier et le dernier caractères de la ligne soient sur les bords de l'écran).

Analyse

Il faut ajouter à la ligne de texte autant de caractères "espace" qu'il lui en manque pour qu'elle compte 4Ø caractères. Ces espaces sont à ajouter en nombre approximativement égal à chaque espace existant entre deux mots dans la ligne initiale.



EXERCICE 3.12

ENTREE : Une phrase ou un mot.

SORTIE : La phrase avec les caractères dans l'ordre inverse et une phrase constituée exactement des mêmes caractères, mais dans un ordre aléatoire.

Exemple : TABLEAU donne UAELBAT en ordre inverse et, par exemple, BLTUAEA en ordre aléatoire.

Analyse

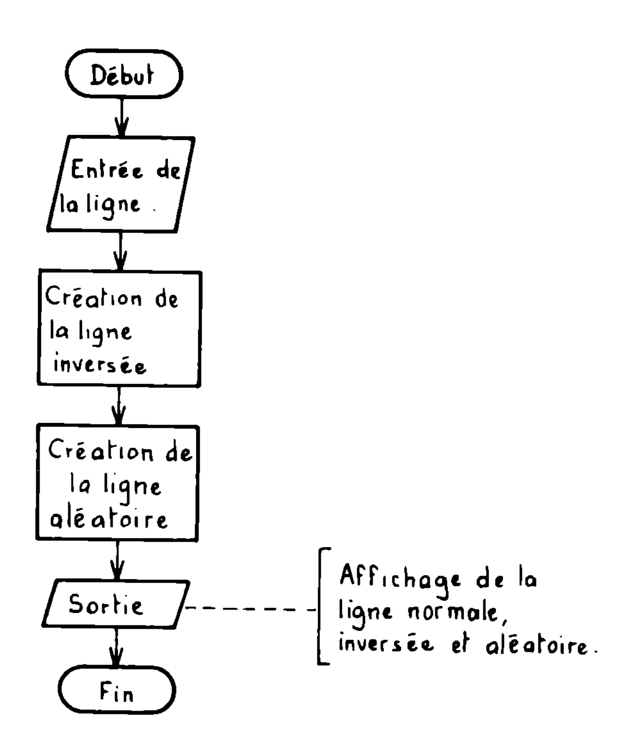
La méthode utilisée pour inverser l'ordre des caractères est la suivante : les caractères de la chaîne à inverser sont ajoutés un à un, en partant de la fin, à une autre chaîne.

Exemple : TABLEAU donne U
puis UA
puis UAE
puis UAEL
puis UAELB
puis UAELBA
et enfin UAELBAT

Pour créer la chaîne aléatoire, un caractère de la chaîne, choisi au hasard, est ajouté à une autre chaîne puis supprimé de la chaîne initiale. Le processus est ensuite recommencé jusqu'à ce que la chaîne initiale soit vide.

Exemple : Avec TABLEAU, on obtiendrait :

```
TALEAU et B
TAEAU  et BL
AEAU   et BLT
AEA    et BLTU
AE     et BLTUA
A      et BLTUAE
et enfin  BLTUAEA
```



EXEMPLE 3.13

PROBLEME : Afficher le code ASCII des touches frappées par l'utilisateur (utiliser la fonction INKEY$).

ENTREE : Une touche du clavier.

SORTIE : Le code ASCII de la touche

Solution

```
10 CLS
20 A$=INKEY$:IF A$="" THEN 20
30 PRINT ASC(A$)
40 GOTO 20
```

Commentaire :

Si ce code est égal à 3 (ce qui correspond à "CTRL" "C"), le programme s'arrête.

PETIT EXERCICE 3.14

SORTIE : Tous les caractères ASCII que peut afficher le TO7.

EXERCICE 3.15

PROBLEME : Transformation de majuscules en minuscules.

ENTREE : Un texte.

SORTIE : Le même texte mais où toutes les majuscules ont été transformées en minuscules.

PETIT EXERCICE 3.16

PROBLEME : Transformer chaque élément d'un tableau de nombres en son pourcentage par rapport à la somme de tous les nombres.

ENTREE : Un tableau de nombres.

SORTIE : Le tableau obtenu en transformant chaque élément en son pourcentage par rapport à la somme de tous les nombres.

EXERCICE 3.17

PROBLEME : Donner le pourcentage de présence de chaque lettre de l'alphabet dans un texte.

ENTREE : Une ou plusieurs phrases de texte écrites en majuscules.

SORTIE : Chaque lettre de l'alphabet suivie de son pourcentage de présence dans les phrases du texte.

PETIT EXERCICE 3.18

PROBLEME : A partir d'une date écrite en chiffres (ex : 15-7-179Ø), remplacer le numéro du mois par son nom.

<u>*Exemple*</u> : "15-7-179Ø" donne "15 JUILLET 179Ø".

ENTREE : Trois nombres correspondant au jour, au mois et à l'année.

SORTIE : Affichage de la date.

EXERCICE 3.19

PROBLEME : Problème inverse du précédent (petit exercice 3.18).

ENTREE : Une date sous la forme d'une chaîne de caractères.

Exemple : "13 JUILLET 1788"

SORTIE : La date écrite en chiffres

Exemple : 13/7/1788

Analyse

La chaîne de caractères représentant la date est découpée en fonction des espaces qu'elle contient. Le numéro du mois est obtenu à partir de la place qu'il occupe dans la liste de tous les mois enregistrée dans une instruction DATA.

EXERCICE 3.20

PROBLEME : Afficher un nombre, entré au clavier, avec des chiffres dessinés au moyen d'étoiles sur une matrice de cinq colonnes et de six lignes.

ENTREE : Un nombre entier positif.

SORTIE : Affichage de ce nombre avec de grands chiffres.

Analyse

Les lignes d'étoiles formant chaque chiffre seront enregistrées sous forme de DATA.

Exemple : le chiffre 4 sera, par cxemple, sous la forme :

```
DATA "    *","   * ","  *  "," * * ","*****","   * "
```

CHAPITRE IV

HISTOGRAMMES, GRAPHIQUES HAUTE ET BASSE RÉSOLUTIONS, TÉLÉCRAN, TRACÉ DE FONCTIONS, ANIMATION

EXEMPLE 4.1

PROBLEME : Afficher le nom et le prénom d'une personne en les soulignant.

ENTREE : Le nom et le prénom d'une personne.

SORTIE : Le nom et le prénom soulignés.

Solution

```
10 CLS
20 INPUT "NOM ";N$
30 INPUT "PRENOM ";P$
35 IF N$="" OR P$="" THEN 20
40 PRINT
50 PRINT TAB(10);N$;"  ";P$
55 PRINT:PRINT:PRINT
60 LINE (10,CSRLIN)-(9+LEN(N$),CSRLIN)"-"
70 LINE (12+LEN(N$),CSRLIN)-(11+LEN(N$)+LEN(P$),CSRLI
N)"-"
80 END
```

Variables

N$ - nom

P$ - prénom

Commentaires

5Ø : L'affichage se fait à partir de la colonne 1Ø de l'écran.

6Ø : Le soulignement du nom se fait à partir de la colonne 1Ø en respectant la longueur du nom.

7∅ : Le soulignement du prénom se fait à partir de la colonne calculée en fonction de la longueur du nom et respecte la longueur du prénom.

PETIT EXERCICE 4.2

PROBLEME : Afficher une ligne et une colonne d'étoiles sur l'écran.

ENTREE : Un nombre L compris entre 0 et 24, un nombre C compris entre 0 et 39.

SORTIE : Affichage d'une ligne de 40 étoiles sur la Lième ligne puis une colonne de 25 étoiles sur la Cième colonne.

EXERCICE 4.3

PROBLEME : Afficher un histogramme à partir de quinze valeurs.

ENTREE : Quinze valeurs comprises entre 0 et 100.

SORTIE : Affichage de l'histogramme correspondant.

Analyse

Il faut faire correspondre à la valeur 100 une bande de hauteur 20, à la valeur 0 une bande de hauteur nulle et aux valeurs intermédiaires une bande de hauteur proportionnelle.

EXERCICE 4.4

PROBLEME : Même programme que le précédent (exercice 4.3), mais pour un nombre de valeurs positives pouvant aller jusqu'à vingt-quatre, sans limitation de grandeur.

ENTREE : Vingt-quatre valeurs positives.

SORTIE : Histogramme correspondant.

Analyse

A la plus grande valeur entrée, il faut faire correspondre le nombre maximum d'étoiles sur une colonne (20).

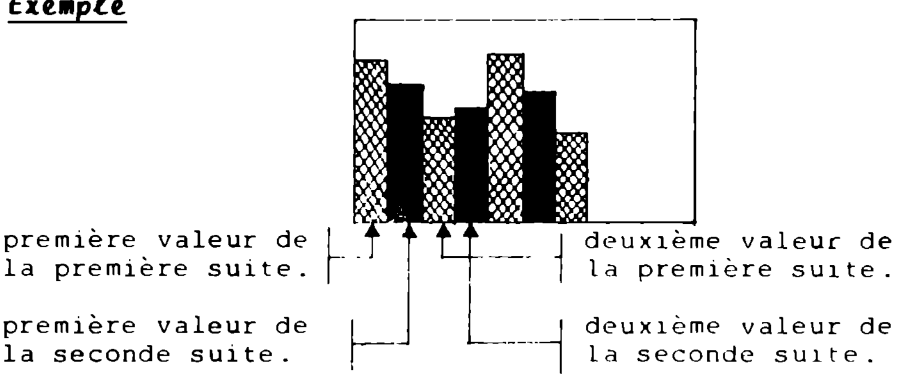
EXERCICE 4.5

PROBLEME : En utilisant la même méthode que celle de l'exercice 4.4, faire l'histogramme comparatif de dix valeurs représentant par exemple les ventes de deux articles sur dix années.

ENTREE : Deux fois dix valeurs.

SORTIE : Histogramme comparatif (utiliser une couleur pour chaque série de valeurs).

Exemple



Analyse

Il faudra, comme précédemment, faire correspondre la plus grande valeur entrée au nombre maximum de "pavés" sur une colonne : 20.

Pour enregistrer les valeurs, il est commode d'utiliser deux tableaux de dix éléments ou un tableau à deux indices.

EXERCICE 4.6

PROBLEME : Dessiner en haute résolution l'histogramme des sommes de 600 lancers de n dés (voir exercices 1.6 à 1.9).

ENTREE : Le nombre de dés à jeter à chaque lancer (entre 1 et 6).

SORTIE : Affichage de l'histogramme en graphique haute résolution, chaque colonne représentant une

des sommes possibles. La hauteur d'une colonne dépend du nombre de fois où la somme correspondante est sortie dans les 600 lancers

Analyse

Pour chaque lancer, la somme des faces sorties est calculée et affichée. A la fin de tous les lancers, l'histogramme de l'ensemble des sommes obtenues est affiché.

EXERCICE 4.7

PROBLEME : Réaliser un quadrillage de l'écran graphique en haute résolution.

ENTREE : Largeur de la maille du quadrillage (nombre de points entre chaque ligne).

SORTIE : Quadrillage de l'écran graphique haute résolution.

Analyse

Pour dessiner un quadrillage, il suffit de tracer des lignes horizontales et des lignes verticales, séparées par la largeur de la maille.

EXERCICE 4.8

PROBLEME : Réalisation d'un télécran graphique.

ENTREE : Les quatre touches de déplacement du curseur.

SORTIE : Dessin sur l'écran par déplacement d'un petit point piloté par les touches de déplacement du curseur.

EXERCICE 4.9

SORTIE : Tracé de la fonction Y = SIN(X) ainsi que des axes des ordonnées et des abscisses avec leurs échelles.

Analyse

Pour tracer la courbe d'une fonction, il faut d'une part préciser la partie de l'écran utilisée et d'autre part les valeurs limites de la fonction, que l'on veut

associer aux bords de la partie de l'écran choisie. Dans notre cas, cette partie sera un rectangle et les valeurs limites de la fonction seront -1 et +1 pour l'ordonnée et 0 et 3*pi pour l'abscisse.

PETIT EXERCICE 4.10

SORTIE : Chute d'un objet (par exemple, une flèche) déclenchée par la pression d'une touche, du haut de l'écran sur une cible horizontale constituée de six cases numérotées. Le programme comptabilise le nombre de coups joués et le score réalisé.

EXERCICE 4.11

SORTIE : Afficher suivant un carré tous les nombres de 1 à 81 puis effacer par une croix tous les nombres non premiers.

Analyse

Un nombre premier est un nombre entier positif qui n'admet comme diviseur que 1 et lui-même.

Si un nombre N est divisible par D, alors N MOD D est égal à Ø. Pour savoir si N est premier, il suffit de vérifier qu'il n'a pas de diviseur compris entre 2 et sa racine carrée.

CHAPITRE V

JEUX GRAPHIQUES, MORPION, PENDU, TIRAGE D'UNE SUITE DE NOMBRES ALÉATOIRES

EXERCICE 5.1 : Jeu du glouton

PROBLEME : Un glouton se dirige du bord gauche au bord droit de l'écran à une vitesse constante. Le but du jeu est de faire parvenir le glouton sur le bord droit en évitant les obstacles répartis aléatoirement sur l'écran. Pour cela le joueur peut faire monter ou descendre le glouton à l'aide des touches de direction du clavier.

ENTREE : Pendant le jeu, les touches de direction (option : un niveau de difficulté).

SORTIE : L'écran parsemé d'obstacles, le glouton se dirigeant du bord gauche au bord droit. Si le glouton rencontre un obstacle, le message "PERDU" s'affiche.

Analyse

Option : le nombre d'obstacles sur l'écran croît en fonction du niveau de difficulté.

EXERCICE 5.2 : Jeu du morpion

PROBLEME : Jeu du morpion pour deux personnes. Sur un damier 3 x 3, chaque joueur remplit à son tour une des cases par son signe, le premier à aligner 3 signes gagne la partie.

ENTREE : Chaque joueur désigne à son tour la case choisie à l'aide du crayon optique (l'utilisation du crayon optique nécessite une intensité

lumineuse soutenue).

SORTIE : Affichage du damier, vérification de la validité des coups et arrêt du jeu dès qu'un joueur a gagné.

Analyse

Il est commode d'utiliser un tableau à deux indices, trois colonnes et trois lignes, pour représenter le damier. Il suffira alors, pour inscrire les coups, de mettre à 1 ou à 2 la case correspondante au coup pour indiquer qu'elle appartient au premier ou au second joueur.

Pour savoir si un des joueurs a gagné, il suffit de passer en revue les lignes, colonnes et diagonales pour voir si l'une d'elles contient trois nombres identiques.

EXERCICE 5.3 : Loto

PROBLEME : Afficher une grille du jeu de loto et simuler un tirage dont le résultat s'affiche sur cette grille.

ENTREE : Une touche est attendue au clavier pour déclencher le tirage.

SORTIE : La grille du loto avec affichage dans une couleur différente des numéros sortis.

Analyse

Lors du tirage des nombres, chaque nouveau nombre tiré est comparé aux nombres déjà tirés et s'il est égal à l'un d'eux, on effectue un nouveau tirage.

EXERCICE 5.4 : Calcul mental

PROBLEME : Faire choisir par l'ordinateur deux nombres et une opération à faire effectuer en temps limité par le joueur.

ENTREE : La réponse du joueur.

SORTIE : Un message en cas de bonne réponse et le résultat exact en cas d'erreur.

Analyse

Utiliser l'instruction INPUTWAIT pour limiter le délai

de réponse imparti au joueur.

EXERCICE 5.5 : Machine à sous

PROBLEME : Réaliser un programme simulant à l'écran une machine à sous.

ENTREE : Une touche pour lancer la machine.

SORTIE : Affichage de la machine et du score de la partie.

Analyse

La machine à sous est composée essentiellement de trois fenêtres dans lesquelles défilent trois symboles différents. Le joueur marque le maximum de points lorsque la machine s'arrête avec l'affichage du symbole gagnant dans les trois fenêtres.

EXERCICE 5.6 : Jeu du pendu

PROBLEME : Trouver un mot caché en proposant des mots de même longueur.

ENTREE : Une touche pour commencer, le nombre de coups choisi et les mots proposés.

SORTIE : Affichage pendant un temps limité de la liste des mots cachés puis de la grille de jeux.

EXERCICE 5.7 : Carnet d'adresses

PROBLEME : Enregistrer sur cassette une liste de noms de personnes avec leur adresse et leur numéro de téléphone, la relire et modifier son contenu.

ENTREE : Les touches correspondant au choix du menu et les noms de personnes avec les renseignements associés.

SORTIE : Affichage d'un menu et du contenu du carnet d'adresses.

Analyse

Les noms avec leurs renseignements sont des chaînes de caractères dont la liste est enregistrée sous forme de fichier. Le programme se décompose en cinq sous-pro-

grammes correspondant à la lecture du fichier, l'affichage de son contenu, l'insertion de nouveaux noms, la suppression de noms anciens et l'enregistrement du fichier sur cassette. Les sous-programmes sont appelés à partir d'un menu.

EXERCICE 5.8 : Mastermind

PROBLEME : L'ordinateur choisit une combinaison de cinq pions de couleurs différentes que le joueur doit découvrir en proposant des combinaisons successives.

ENTREE : Les touches de démarrage, le nombre d'essais demandés et les combinaisons proposées.

SORTIE : Un écran donne les règles du jeu et la réponse de l'ordinateur à l'essai proposé indique le nombre de pions bien placés et de pions mal placés.

Analyse

La combinaison à découvrir est créée de façon aléatoire sous la forme d'un tableau de cinq lettres représentant chacune une couleur. L'essai proposé est constitué d'un tableau de même structure. Le problème consiste à comparer un à un les éléments de ces deux tableaux.

EXERCICE 5.9 : Jeu de NIMM

PROBLEME : L'écran affiche les allumettes alignées que les joueurs retirent en précisant la ligne et le nombre d'allumettes.

ENTREE : Les coups joués (ligne et nombre) par chacun des joueurs.

SORTIE : Affichage des allumettes et nom du vainqueur.

2e PARTIE :

PROGRAMMES COMMENTÉS
SOLUTION DES EXERCICES

EXERCICES POUR TO7

EXERCICE 1.2

Programme

```
10 CLS
20 INPUT "QUEL EST TON PRENOM";P$
30 INPUT "QUELLE EST TON ANNEE DE NAISSANCE";A
40 PRINT
50 PRINT "EH BIEN BONJOUR ";P$
60 PRINT "JE NE REPETERAI A PERSONNE QUE TU AS      AUJOURD'HUI";1984-A;"ANS."
70 END
```

Variables

A - année de naissance

P$ - prénom

Commentaires

40 : Saute une ligne pour faciliter la lecture.

60 : 1984 - A donne l'âge (remplacer éventuellement 1984 par l'année actuelle).

EXERCICE 1.3

Programme

```
10 CLS
20 INPUT "QUEL NOMBRE";N
30 INPUT "QUEL TAUX DE POURCENTAGE";P
40 PRINT
50 DN=N*P/100
60 PRINT "VARIATION :";DN
70 PRINT "VALEUR FINALE :";N+DN
80 END
```

Variables

N - valeur numérique

P - taux de pourcentage

DN - pourcentage

Commentaires

50 : Calcul de la variation

70 : Affichage de la valeur numérique finale.

EXERCICE 1.4

Programme

```
10 CLS
20 INPUT "QUEL EST VOTRE PRENOM";P$
30 INPUT "COMBIEN DEPENSEZ-VOUS QUOTIDIENNEMENT   POU
R VOTRE NOURRITURE";D
40 DA=D*365*1.07
50 CLS
60 PRINT "HE BIEN ";P$
70 PRINT : PRINT
80 PRINT "VOUS DEPENSEZ A PEU PRES";DA;"FRANCS DE NOU
RRITURE PAR AN."
90 LOCATE 7,10
100 PRINT "MOI JE DEPENSE 0 CENTIME !!"
110 LOCATE 0,23
120 END
```

Variables

D - dépense quotidienne

DA - dépense annuelle

P$ - prénom

Commentaires

40 : Calcul de la dépense annuelle

110 : Positionnement du curseur à la ligne 23, pour éviter l'apparition de OK sous la dernière ligne affichée.

EXERCICE 1.5

Programme

```
10 CLS
20 REM ENTREE DU NOMBRE A DECOUVRIR
30 INPUT "QUEL  NOMBRE  ENTIER  ENTRE  0 ET 1000  VOU
LEZ-VOUS CACHER ";N
40 IF N-INT(N)<>0 OR N<0 OR N>1000 THEN 30
50 REM DEBUT DU JEU
60 CLS
70 PRINT
80 INPUT "ESSAI ";E
90 IF E>N THEN PRINT "TROP GRAND":GOTO 70
100 IF E<N THEN PRINT "TROP PETIT":GOTO 70
110 PRINT "BRAVO, VOUS AVEZ TROUVE !!"
120 END
```

Variables

N - nombre à découvrir

E - essai

Commentaires

4Ø : Le nombre à découvrir est testé sur les trois conditions suivantes :
- Entier
- Supérieur ou égal à Ø
- Inférieur ou égal à Ø

S'il est inférieur à Ø ou supérieur à 1ØØØ un autre nombre est demandé.

7Ø : Passe une ligne entre chaque essai.

11Ø : Si les résultats des deux tests précédents sont faux (les deux nombres étant égaux) la ligne 11Ø est exécutée.

EXERCICE 1.7

Programme

```
10 CLS
20 INPUT "NOMBRE DE DES LANCES (1 A 8)  (0 POUR ARR
ETER) ";N
30 PRINT
40 IF N=0 THEN END
50 IF N<1 OR N>8 THEN 20
60 S=0
70 FOR I=1 TO N
80 S=S+INT(6*RND)+1
90 NEXT I
100 PRINT "SOMME = ";S
110 PRINT:PRINT
120 GOTO 20
```

Variables

I - compteur

N - nombre de dés à lancer

S - somme des nombres sortis pour un lancer.

Commentaires

4Ø : Si N= Ø le programme s'arrête.

5Ø : Cette ligne doit se trouver après la ligne 4Ø de préférence.

sinon le programme ne s'arrêterait pas pour N=Ø.

Remarque : Pour calculer la somme du lancer de N dés, la variable S est initialisée à Ø en 6Ø, puis, dans la boucle 7Ø-9Ø, augmentée de la valeur obtenue à chacun des N lancers.

EXERCICE 1.8

Programme

```
10 CLS
20 REM INITIALISATION DES VARIABLES
30 N=INT(1000*RND)+1
40 C=0
50 REM DEBUT DU JEU
60 C=C+1
70 PRINT:INPUT "ESSAI ";E
80 IF E>N THEN PRINT "TROP GRAND":GOTO 50
90 IF E<N THEN PRINT "TROP PETIT":GOTO 50
100 PRINT:PRINT
110 PRINT "BRAVO VOUS AVEZ TROUVE EN ";C;" COUPS."
120 PRINT:PRINT
130 END
```

Variables

C - compteur des essais

N - nombre à découvrir

E - numéro de l'essai

Commentaires

3Ø : Un nombre entier est choisi au hasard entre Ø et 1ØØØ.

6Ø : Le compteur C est augmenté de 1 (il est "incrémenté"). Ne pas oublier de le mettre à Ø en début de programme (ligne 4Ø).

1ØØ : Deux lignes blanches sont passées. Cette ligne, et les suivantes, ne seront exécutées que si les résultats des deux tests précédents sont faux.

11Ø : C contient le nombre d'essais.

Remarque : Vous constatez que ce programme choisit toujours le même nombre à découvrir : en effet, l'exécution de l'instruction RUN initialise la série des nombres aléatoires fournis par l'instruction RND. L'exercice 1.9 donne une des solutions pour initialiser de manière

aléatoire l'instruction RND.

EXERCICE 1.9

Programme

```
10 CLS
20 REM INITIALISATION DES VARIABLES
30 LOCATE 10,10,0
40 PRINT "APPUYER SUR UNE TOUCHE QUELCONQUE POUR  COM
MENCER"
50 FOR T=0 TO 10000
60 A$=INKEY$
70 IF A$<>"" THEN 100
80 NEXT T
90 REM
100 CLS
110 N=INT(1000*RND(-T))+1
120 C=0
130 REM DEBUT DU JEU
140 C=C+1
150 PRINT:INPUT "ESSAI ";E
160 IF E>N THEN PRINT " TROP GRAND"   GOTO 130
170 IF E<N THEN PRINT " TROP PETIT"   GOTO 130
180 PRINT:PRINT
190 PRINT "BRAVO VOUS AVEZ TROUVE EN ";C;" COUP(S)."
200 PRINT:PRINT
210 END
```

Variables

T - compteur de boucle

A$ - touche frappée au clavier

C - compteur des essais

N - nombre à découvrir

E - numéro de l'essai

Commentaires

50-80 : La sortie de la boucle s'effectue au moment où une touche est appuyée et donc la valeur de T est fonction du temps de réaction du joueur.

110 : La valeur de T est utilisée pour initialiser la fonction RND.

EXERCICE 2.2

Programme

```
10 CLS
20 REM ENTREE DES DONNEES
30 INPUT "PRIX H.T. ";P
40 INPUT "CODE T.V.A. (A:5.5% B:18.6% C:33%) ";CT$
50 REM
60 REM RECHERCHE DU TAUX DE T.V.A.
70 IF CT$="A" THEN T=5.5 :GOTO 110
80 IF CT$="B" THEN T=18.6 :GOTO 110
90 IF CT$="C" THEN T=33 :GOTO 110
100 GOTO 40
110 REM
120 REM CALCUL ET AFFICHAGE
130 PRINT
140 PT=P+P*T/100
150 PT=INT(PT*100)/100
160 PRINT "PRIX T.T.C.   ";PT
170 PRINT:PRINT
180 END
```

Variables

P - prix hors-taxe

PT - prix TTC

T - taux TVA

CT$- code du taux TVA

Commentaires

40 : Les codes à utiliser sont rappelés dans le texte de l'instruction INPUT.

70-90: Le taux de TVA est affecté à la variable T.

100 : Si les trois tests précédents donnent un résultat négatif, cela signifie que le code entré n'est pas valable et un autre code est alors demandé.

140 : Calcul du prix TTC.

150 : Arrondi à deux décimales : le chiffre est multiplié par 100, ainsi les deux décimales que l'on désire conserver se trouvent à gauche du point décimal (ex : 6.6734 x 100 donne 667.34). Ensuite la valeur entière du nombre est prise, supprimant les décimales que l'on ne veut pas conserver (ex : INT (667.34) donne 667). Puis pour

retrouver la valeur initiale le chiffre est divisé par 1ØØ (ex : 667/1ØØ donne 6.67, c'est-à-dire 6.6734 arrondi à deux décimales).

Remarque : La valeur est arrondie par défaut avec une erreur comprise entre Ø et 1 sur la dernière décimale. Pour obtenir un arrondi correspondant à la valeur entière la plus proche, il suffit d'écrire :

150 PT=INT(PT*1ØØ+.5)/1ØØ

Lorsque l'arrondi n'est nécessaire qu'à l'affichage, il est plus simple d'utiliser l'instruction PRINT USING (voir exercice 2.3)

EXERCICE 2.3

Programme

```
1Ø CLS
2Ø DIM P(1Ø)
3Ø REM
4Ø REM ENTREE DES DONNES
5Ø FOR I=1 TO 1Ø
6Ø PRINT "PRIX ";I;
7Ø INPUT P(I)
8Ø NEXT I
9Ø PRINT
1ØØ INPUT "CODE DU TAUX DE T.V.A. (A:5.5% B:18.6% C:3
3%) ";CT$
11Ø REM
12Ø REM RECHERCHE DU TAUX DE T.V.A
13Ø IF CT$="A" THEN T=5.5 :GOTO 18Ø
14Ø IF CT$="B" THEN T=18.6 :GOTO 18Ø
15Ø IF CT$="C" THEN T=33 :GOTO 18Ø
16Ø GOTO 9Ø
17Ø REM
18Ø REM CALCUL
19Ø FOR I=1 TO 1Ø
2ØØ P(I)=P(I)+P(I)*T/1ØØ
21Ø NEXT I
22Ø REM
23Ø REM AFFICHAGE AVEC ALIGNEMENT
24Ø PRINT
25Ø FOR I=1 TO 1Ø
26Ø PRINT "PRIX ";I;TAB(1Ø)
27Ø PRINT USING "######.##";P(I)
28Ø NEXT I
29Ø END
```

Variables

I - compteur de boucle

T - taux TVA

CT$ - code du taux TVA

P(10) - tableau des prix

Commentaires

2Ø : Dimensionnement du tableau

5Ø-8Ø : Boucle d'entrée des prix. Les dix prix sont demandés un par un, le numéro du prix à entrer est affiché à la ligne 6Ø (voir exemple 2.1).

19Ø-21Ø : Boucle de calcul des prix TTC. Les prix TTC remplacent les prix hors-taxe dans le tableau.

25Ø-28Ø : Boucle d'affichage.

27Ø : L'instruction PRINT USING permet l'alignement sur le point décimal et l'arrondi (il est ici prévu pour des nombres inférieurs à un million).

EXERCICE 2.4

Programme

1ère méthode

```
1Ø CLS
2Ø DIM L$(2Ø)
3Ø READ N
4Ø FOR I=1 TO N
5Ø READ L$(I)
6Ø NEXT I
7Ø FOR I=1 TO N:PRINT L$(I):NEXT I
8Ø PRINT:PRINT
9Ø END
5ØØ DATA 7
51Ø DATA TO7,TABLE,VOITURE,ARBRE,MAISON,CHAISE,OISEAU
```

2ème méthode

```
10 CLS
20 DIM L$(20)
30 FOR I=1 TO 20
40 READ L$(I)
50 IF L$(I)<>"FIN" THEN NEXT I
60 N=I-1
70 FOR I=1 TO N:PRINT L$(I):NEXT I
80 PRINT:PRINT
90 END
500 DATA TO7,TABLE,VOITURE,ARBRE,MAISON,CHAISE,OISEAU
1000 DATA FIN
```

3ème méthode

```
10 DIM L$(20)
20 ON ERROR GOTO 1000
30 FOR I=1 TO 100
40 READ L$(I)
50 NEXT I
60 CLS
70 FOR I=1 TO N:PRINT L$(I):NEXT I
80 PRINT:PRINT
90 END
500 DATA TO7,TABLE,VOITURE,ARBRE,MAISON,CHAISE,OISEAU
1000 N=I-1
1010 RESUME 70
```

Variables

I - compteur

N - nombre de mots contenus dans L$(2Ø)

L$(2Ø) - tableau des mots

Commentaires

Remarque : Les trois méthodes donnent des résultats similaires.

- Pour que la *première méthode* soit possible, il faut connaître le nombre d'éléments à lire (ici, celui-ci est lu par l'instruction READ N à laquelle correspond l'instruction DATA 7 en ligne 5ØØ).
- Pour pouvoir utiliser la *deuxième méthode* il faut qu'une des données indique la fin de la lecture (ligne 1ØØØ DATA FIN). La ligne 6Ø de la deuxième méthode a été rajoutée pour qu'à la fin de l'exécution du programme la variable N contienne le nombre de mots dans L$.

La troisième méthode ne nécessite ni de connaître le nombre d'éléments de la liste des données ni de réserver un mot pour en indiquer la fin. C'est l'erreur provoquée par le défaut de données à lire par l'instruction READ qui est utilisée pour sortir de la boucle de lecture. Cette méthode est à utiliser avec précaution et uniquement comme raccourci pour le test d'un programme par exemple.

EXERCICE 2.5

Programme

```
10 CLS
20 DIM N$(20),P$(20)
30 REM
40 REM LECTURE DES DONNEES
50 READ N
60 FOR I=1 TO N
70 READ N$(I),P$(I)
80 NEXT I
90 REM
100 INPUT "NOM A CHERCHER ";N$
110 INPUT "PRENOM ",P$
120 PRINT
130 REM
140 REM RECHERCHE
150 FOR I=1 TO N
160 IF N$=N$(I) AND P$=P$(I) THEN PRINT P$;" ";N$;" E
ST DANS LE FICHIER.":GOTO 190
170 NEXT I
180 PRINT P$;" ";N$;" N'EST PAS DANS LE FICHIER."
190 PRINT:PRINT
200 END
500 DATA 5
510 DATA MARTIN,ALBERT,PERRIN,ADELAIDE,BARON,JEAN
520 DATA DURAND,GEORGES,MARTEAU,JACQUES
```

Variables

I - compteur

N - nombre de personnes

N$ - nom à chercher

P$ - prénom à chercher

N$(20) - tableau des noms

P$(20) - tableau des prénoms

Commentaires

60-80 : Lecture des noms et prénoms avec la première méthode de l'exercice 2.4.

160 : Si les nom et prénom sont ceux de la personne cherchée, le programme affiche un message puis s'arrête. Sinon, la personne suivante dans la liste est testée.

180 : Si toutes les personnes ont été testées et qu'aucune n'est celle demandée, le programme l'affiche.

Remarque : N$ et N$(n) sont deux variables distinctes, la première est la variable alphanumérique N$, la seconde est le nième élément du tableau alphanumérique N$.

EXERCICE 2.6

Programme

```
10 CLS
20 REM
30 REM LECTURE DU NOMBRE DE PERSONNES
40 READ N
50 REM
60 REM ENTREE DU NOM ET DU NUMERO
70 INPUT "NOM OU NUMERO A RECHERCHER (TAPEZ  FIN  POU
R  ARRETER) ";R$
80 IF R$="FIN" THEN END
90 R=LEN(R$)
100 REM
110 REM RECHERCHE
120 F=0
130 FOR I=1 TO N
140 READ D1$,D2$,D3$
150 IF R$=MID$(D1$,1,R) OR R$=MID$(D3$,1,R) THEN F=1
:PRINT D1$,"  ";D2$,"....";D3$
160 NEXT I
170 IF F=0 THEN PRINT R$;" N'EXISTE PAS DANS LA LISTE
."
180 REM
190 RESTORE:PRINT
200 GOTO 30
490 DATA 6
500 DATA MARTIN,ALBERT,275 98 98,PERRIN,ADELAIDE,67 7
6 01
```

```
510 DATA DURAND,GEORGES,23 65 72,MARTEAU,JACQUES,575
76 27
520 DATA MARTIN,JEAN,345 82 54,MOORE,RICK,19 1 714 75
5 5004
```

Variables

I - compteur

N - nombre de personnes dans le fichier

R - nombre de caractères dans R$

F - drapeau (flag) : 0 si la personne cherchée n'est pas dans le fichier

R$ - nom ou numéro à rechercher

D1$ - nom de la Ième personne de la liste

D2$ - prénom de la Ième personne

D3$ - numéro de la Ième personne

Commentaires

80 : Le programme bouclant en 200 (GOTO 30) est interrompu lorsque la réponse FIN est donnée à la ligne 70.

150 : Si le nom ou le numéro de téléphone correspondent, alors la personne est affichée et le drapeau est levé (F=1). La recherche continue néanmoins car plusieurs personnes peuvent avoir le même nom, ou le même numéro de téléphone.

170 : Si F contient 0, cela indique qu'aucune personne du fichier n'est celle demandée.

190 : L'instruction RESTORE permet de relire les données en DATA à partir du début (y compris le nombre de personnes).

Remarque : La comparaison s'effectue entre la chaîne entrée (R$) et les R premiers caractères du nom ou du numéro. Ainsi, il est possible de faire afficher la liste des personnes dont le nom commence par "M" en tapant cette lettre seule lors de la demande du nom ou du numéro (ligne 70).

EXERCICE 2.7

Programme

```
10 DIM D(100)
20 CLS
30 INPUT "ENTRER COMBIEN DE NOMBRES (DE 1 A 100) ";N
40 PRINT
50 IF N<1 OR N>100 THEN 30
60 REM
70 REM ENTREE DES DONNEES
80 FOR I=1 TO N
90 PRINT "NOMBRE ";I;
100 INPUT D(I)
110 NEXT I
120 REM
130 REM RECHERCHE DU MIN-MAX
140 MI=D(1):MA=D(1)
150 FOR I=1 TO N
160 IF MI>D(I) THEN MI=D(I)
170 IF MA<D(I) THEN MA=D(I)
180 NEXT I
190 REM
200 REM AFFICHAGE
210 PRINT:PRINT
220 PRINT "MAXIMUM   ";MA
230 PRINT "MINIMUM   ";MI
240 PRINT
250 END
```

Variables

I - compteur

N - nombre d'éléments du tableau D(1ØØ)

MI - plus petit nombre

MA - plus grand nombre

D(1ØØ) - tableau des nombres

Commentaires

15Ø-18Ø : Tous les éléments du tableau sont comparés à MI et MA. Si l'élément est plus grand que MA alors il est mis dans MA. S'il est plus petit que MI, il est mis dans MI.

Remarque : Il serait possible d'écrire :

15Ø FOR I=2 TO N

car il est inutile de comparer le premier élément du

tableau à MI et MA qui le contiennent déjà. Néanmoins, cette solution provoquerait une erreur pour le cas où le tableau ne contiendrait qu'un élément (N=1).

EXERCICE 2.8

Programme

```
10 DIM N$(1,20)
20 CLS
30 INPUT "COMBIEN DE PERSONNES (DE 1 A 20) ";N
40 IF N<1 OR N>20 THEN 30
50 PRINT
60 REM
70 REM ENTREE DES DONNEES
80 FOR I=1 TO N
90 PRINT "PERSONNE ";I;":"
100 INPUT "NOM ";N$(0,I)
110 INPUT "PRENOM ";N$(1,I)
120 PRINT
130 NEXT I
140 REM
150 REM RECHERCHE
160 M=1
170 FOR I=1 TO N
180 IF N$(0,I)=N$(0,M) AND N$(1,I)<N$(1,M) THEN M=I
190 IF N$(0,I)<N$(0,M) THEN M=I
200 NEXT I
210 REM
220 REM AFFICHAGE
230 PRINT
240 PRINT "LA   PREMIERE   PERSONNE   PAR  ORDRE   AL
PHABETIQUE   EST"
250 PRINT "CELLE D'INDICE ";M;":   ";N$(1,M);"   ";N$(0
,M)
260 PRINT
270 END
```

Variables

I - compteur

N - nombre de personnes

M - indice de la première personne dans l'ordre alphabétique

N$(1,2Ø) - tableau des noms et prénoms (N$(Ø,x) - noms
N$(1,x) - prénoms

Commentaires

9Ø : Le numéro de la personne à entrer est affiché.

18Ø : Si les noms sont identiques, l'ordre des prénoms est alors pris en compte.

Remarque : Un tableau à deux indices est utilisé. Le premier indice indique s'il s'agit d'un nom (Ø) ou d'un prénom (1). Le deuxième indice indique le rang dans la liste.

	Ø	1
1	nom 1	prénom 1
2	nom 2	prénom 2
3	.	.
4		.
5	.	.
6	.	.

NS(1,2Ø)

Il est possible de regrouper les tests des lignes 18Ø et 19Ø en un seul : 18Ø IF N$(Ø,I) < N$(Ø,M) OR < N$ (Ø,I) = N$(Ø,M) AND N$(1,I) > N$(1,M) > THEN M=I

EXERCICE 2.9

Programme

1ère méthode

```
10 DIM TA(50),TB(50)
20 REM ENTREE DES DONNEES
30 READ N1
40 FOR I=1 TO N1:READTA(I):NEXT I
50 REM
60 REM CLASSEMENT
70 N2=0
80 REM RECHERCHE DE L'ELEMENT LE PLUS GRAND DU PREMIE
R TABLEAU
90 N2=N2+1
100 M=1
110 FOR I=1 TO N1
120 IF TA(I)>TA(M) THEN M=I
130 NEXT I
```

```
140 REM
150 REM AJOUT DANS LE SECOND TABLEAU
160 TB(N2)=TA(M)
170 TA(M)=-1
180 IF N2<N1 THEN 90
190 REM
200 REM AFFICHAGE
210 FOR I=1 TO N2:PRINT TB(I);:NEXT I
220 PRINT
230 END
500 DATA 12
510 DATA 3,10,15,1000,1,5,32,0,15,12,26,38
```

2ème méthode

```
10 DIM TA(50),TB(50)
20 REM ENTREE DES DONNEES
30 READ N1
40 FOR I=1 TO N1:READ TA(I):NEXT I
50 REM
60 REM CLASSEMENT
70 N2=0
80 REM RECHERCHE DE L'ELEMENT LE PLUS GRAND DU PREMIE
R TABLEAU
90 N2=N2+1
100 M=1
110 FOR I=1 TO N1
120 IF TA(I)>TA(M) THEN M=I
130 NEXT I
140 REM
150 REM ECHANGE DES ELEMENTS
160 TB(N2)=TA(M)
170 IF M<N1 THEN FOR I=M+1 TO N1:TA(I-1)=TA(I):NEXT I
180 N1=N1-1
190 IF N1>0 THEN 90
200 REM
210 REM AFFICHAGE
220 FOR I=1 TO N2:PRINT TB(I);:NEXT I
230 PRINT
240 END
500 DATA 12
510 DATA 3,10,15,1000,1,5,32,0,15,12,26,38
```

Variables

I - compteur

N1 - nombre d'éléments du premier tableau

N2 - nombre d'éléments du second tableau

M - indice de l'élément le plus grand du premier tableau

TA(50) - premier tableau : nombres à classer

TB(50) - second tableau : nombres classés.

Commentaires

30-40 : Lecture des données (première méthode de l'exercice 2.4).

110-130 : Recherche de l'élément le plus grand du premier tableau (voir l'exercice 2.7).

Remarque : Le nombre en DATA de la ligne 500 indique le nombre d'éléments à lire alors que ceux de la ligne 510 sont les éléments à classer.

EXERCICE 2.10

Programme

```
10 DIM N$(1,20)
20 REM
30 REM ENTREE DES DONNEES
40 CLS
50 INPUT "COMBIEN DE PERSONNES (DE 1 A 20) ",N
60 IF N<1 OR N>20 THEN 30
70 FOR I=1 TO N
80 PRINT "PERSONNE ";I;":"
90 INPUT "NOM ";N$(0,I)
100 INPUT "PRENOM ";N$(1,I)
110 PRINT:NEXTI
120 REM
130 REM CLASSEMENT
140 P=1
150 M=P
160 FOR I=P TO N
170 IF N$(0,I)<N$(0,M) OR (N$(0,I)=N$(0,M) AND N$(1,I
)<N$(1,M)) THEN M=I
180 NEXT I
190 REM
200 N$=N$(0,P):P$=N$(1,P)
210 N$(0,P)=N$(0,M):N$(1,P)=N$(1,M)
220 N$(0,M)=N$:N$(1,M)=P$
230 P=P+1:IF P<N THEN 150
240 REM
250 REM AFFICHAGE
260 FOR I=1 TO N:PRINT N$(0,I),N$(1,I):NEXT I
270 END
```

Variables

I - compteur

N - nombre de personnes

P - indice à partir duquel les personnes doivent être classées

M - indice de la première personne par ordre alphabétique

N$,P$ - variables

N$(1,20) - tableau des noms et prénoms (N$(0,x) - noms, N$(1,x) - prénoms).

Commentaires

150-180 : Recherche de la première personne par ordre alphabétique parmi celle d'indice supérieur ou égal à P.

200-220 : Permutation de la Pième personne avec la Mième.

Remarque : Pour permuter deux variables, par exemple N1 et N2, il faut utiliser une troisième variable, par exemple A, et exécuter les opérations suivantes :

```
A  = N1 ; recopier N1 dans A
N1 = N2 ; recopier N2 dans N1
N2 = A  ; recopier A dans N2
```

EXERCICE 2.11

Programme

```
10 DIM N$(1,20)
20 REM
30 REM ENTREE DE LA LISTE DE PERSONNES
40 READ N
50 FOR I=1 TO N:READ N$(0,I),N$(1,I):NEXT I
60 REM
70 REM ENTREE DE LA PERSONNE A AJOUTER
80 CLS
90 INPUT "NOM A AJOUTER (TAPER FIN POUR ARRETER)";N$
100 IF N$="FIN" OR N=20 THEN END
110 INPUT "PRENOM ";P$
120 REM
130 REM RECHERCHE
140 FOR I=1 TO N
150 IF N$<N$(0,I) OR(N$=N$(0,I) AND P$<N$(1,I)) THEN
    220
```

```
160 NEXT I
170 REM
180 REM AJOUT A LA FIN DU TABLEAU
190 N$(0,N+1)=N$:N$(1,N+1)=P$
200 GOTO 270
210 REM
220 REM DECALAGE ET AJOUT
230 FOR J=N TO I STEP -1
240 N$(0,J+1)=N$(0,J):N$(1,J+1)=N$(1,J)
250 NEXT J
260 N$(0,I)=N$:N$(1,I)=P$
270 N=N+1
280 REM
290 REM AFFICHAGE
300 CLS
310 FOR I=1 TO N:PRINT N$(0,I),N$(1,I):NEXT I
320 PRINT:PRINT:GOTO 90
500 DATA 5
510 DATA ALRIC,JEAN,DURAND,GEORGES,MARTEAU,JACQUES
520 DATA PERRIN,ADELAIDE,PERRIN,PAUL
```

Variables

I,J - compteurs

N - nombre de personnes

N$,P$ - nom et prénom de la personne à insérer

N$(1,20) - tableau des noms et prénoms (N$(0,x) - noms, N$(1,x) - prénoms).

Commentaires

90 et 100 : L'utilisateur peut insérer plusieurs personnes (le programme boucle en 320) et doit taper FIN pour arrêter.

100 : Si le tableau contient 20 personnes, il est impossible d'en insérer une autre.

140-160 : Si la position d'insertion dans le tableau n'est pas la dernière, alors le programme continuera en 220 et I contiendra cette position. Si c'est la dernière alors la boucle se termine et le programme continuera en 170.

270 : Ne pas oublier d'incrémenter N (le tableau contient une personne de plus).

EXERCICE 2.12

Programme

```
10 DIM N$(20),V$(20),C$(20)
20 REM LECTURE DES PHRASES
30 READ N
40 FOR I=1 TO N:READ N$(I),V$(I),C$(I):NEXT I
50 REM AFFICHAGE DES PHRASES
60 CLS
70 FOR I=1 TO 14
80 PRINT N$(N*RND+.5);" ";V$(N*RND+.5);" ";C$(N*RND+.
5)
90 NEXT I
100 PRINT:PRINT
110 REM ATTENTE POUR LA SUITE
120 PRINT "TAPER  SUR  'ENTREE'  POUR  CONTINUER,  SU
R A POUR ARRETER."
130 A$=INKEY$:IF A$="" THEN 130
140 IF A$<> "A" THEN 50
150 END
500 DATA 6
510 DATA L'ORDINATEUR,EST,UNE MACHINE
520 DATA L'HOMME,EST,UN ETRE VIVANT
530 DATA LE CHIEN,DORT SUR,LE LIT
540 DATA LE MUSEE,OUVRE,LE SOIR
550 DATA LE TRAIN,ROULE SUR,LA VOIE FERREE
560 DATA LA FUSEE,S'EST ENVOLEE VERS,LA LUNE
```

Variables

I - compteur

N - nombre d'éléments dans N$(20), V$(20) et C$(20)

A$ - touche frappée au clavier

N$(20) - noms

V$(20) - verbes

C$(20) - compléments

Commentaires

40 et 510-560 : Les différents éléments des phrases sont séparés par une virgule, pour permettre la lecture de chacun d'entre eux.

80 : Affichage d'une phrase. Chacun des éléments la constituant est choisi au hasard.

13Ø-14Ø : Le programme continuera à afficher des phrases jusqu'à ce que l'utilisateur tape "A".

Remarque : Pour permettre la lecture des phrases, celles-ci sont affichées par groupe de 14, une pression de touche étant attendue au clavier entre chaque affichage.

PETIT EXERCICE 3.1

Programme

```
10 CLS
20 INPUT "TAPEZ UNE LIGNE ";A$
30 I=1
40 IF MID$(A$,I,1)=" " THEN A$=LEFT$(A$,I-1)+MID$(A$,
I+1):GOTO 40
50 I=I+1:IF I<=LEN(A$) THEN 40
60 PRINT "(";A$;")"
70 END
```

Variables

I - compteur

A$ - ligne à traiter

Commentaires

Tous les caractères de la chaîne sont comparés un par un à l'espace (ligne 4Ø). Si le Ième caractère est un espace il est supprimé de la chaîne, le programme revient alors en 4Ø car le Ième caractère a changé. Sinon I est incrémenté. Le processus recommence jusqu'à ce que I soit supérieur à la longueur de la chaîne.

4Ø : Pour supprimer le Ième caractère de A$, il suffit de mettre dans A$ tous les caractères précédant le Ième (LEFT$(A$,I-1)), ainsi que ceux qui le suivent (MID$(A$,I+1)).

Remarque : Il est déconseillé d'utiliser la boucle FOR-NEXT.

```
30 FOR I=1 TO LEN (A$)
40........
50 NEXT I
```

car LEN (A$) est évalué comme la longueur initiale de la chaîne.

PETIT EXERCICE 3.2

Programme

```
10 CLS
20 INPUT "TAPEZ UN MOT ";A$
30 IF LEN(A$)<2 THEN 70
40 FOR I=LEN(A$)-1 TO 1 STEP -1
50 A$=LEFT$(A$,I)+" "+MID$(A$,I+1)
60 NEXT I
70 PRINT A$
80 PRINT:PRINT
90 END
```

Variables

I - compteur

A$ - mot à traiter

Commentaires

30 : Il est impossible d'insérer des espaces si la longueur du mot est 1 ou 0.

40 : LEN(A$) est évalué comme la longueur initiale de A$.

Remarque : Les espaces sont ajoutés à partir de la fin du mot. Pour commencer par le début il faudrait écrire

```
50 FOR I=1 TO LEN(A$)*2-2 STEP 2
60.......
70 NEXT I
```

PETIT EXERCICE 3.3

Programme

```
10 CLS
20 LINE INPUT "TAPEZ LE TEXTE " P$
30 PRINT
40 PRINT
50 PRINT P$
60 P$=MID$(P$,2)
70 IF LEN(P$)>0 THEN 50
80 END
```

Variables

PØ - texte à traiter

Commentaires

6Ø : MIDØ (PØ,2) donne tous les caractères de PØ sauf le premier.

7Ø : S'il reste des caractères dans PØ, alors le programme boucle en 5Ø.

EXERCICE 3.4

Programme

```
10 CLS
20 LINE INPUT "TAPEZ UNE LIGNE :";L$
30 PRINT L$
40 IF LEN(L$)<2 THEN 90
50 FOR I=1 TO LEN(L$)/2
60 L$=MID$(L$,2,LEN(L$)-2)
70 PRINT TAB(I);L$
80 NEXT I
90 END
```

Variables

I - compteur

LØ - ligne à traiter

Commentaires

6Ø : Le premier et le dernier caractères ae LØ lui sont enlevés

7Ø : L'affichage à la position I permet de réaliser la tabulation demandée.

EXERCICE 3.5

Programme

```
10 CLS
20 LINE INPUT "TAPEZ LA PHRASE :";P$
30 REM
40 REM DEBUT DE LA BOUCLE
50 PRINT P$
60 FOR I=1 TO LEN(P$)
70 IF MID$(P$,I,1)=" " THEN P$=MID$(P$,I+1):GOTO 50
```

```
80 NEXT I
90 PRINT
100 END
```

Variables

I - compteur

P$ - phrase à traiter

Commentaires

70 : Si le Ième caractère est un espace, alors tous les caractères le précédant et lui-même sont supprimés de P$.

EXERCICE 3.7

Programme

1ère méthode

```
10 CLS
15 A$="........................................"
20 INPUT "LIGNE (MAX:39 CARACTERES)";L$
30 IF LEN(L$)>39 THEN 20
40 REM
50 REM CREATION DE LA LIGNE DE 40 CARACTERES
60 L$=L$+RIGHT$(A$,40-LEN(L$))
70 CLS
80 REM DEFILEMENT
90 LOCATE 0,10,0
100 PRINT L$
110 L$=RIGHT$(L$,39)+LEFT$(L$,1)
120 FOR I=1 TO 40:NEXT I
130 GOTO 90
```

Variables

I - compteur

L$ - ligne à faire défiler

A$ - ligne de points

Commentaires

60 : La ligne est complétée par des points pour obtenir 40 caractères au total.

100 : Affichage de L$ sur la 10ème ligne de l'écran

11Ø : Le premier caractère de la ligne est placé en dernière position.

12Ø : boucle d'attente.

Programme

2ème méthode

```
10 CLS
20 INPUT "TEXTE (MAX:39 CARACTERES)",L$
30 IF LEN(L$)>39 THEN 20
40 L$=L$+" "
50 REM DEFILEMENT
60 FOR I=40 TO 1 STEP -1
70 LOCATE 0,10,0
80 PRINT MID$(L$,41-I);TAB(I);LEFT$(L$,40-I)
90 FOR J=1 TO 20 : NEXT J
100 NEXT I
110 GOTO 60
```

Variables

I,J - compteurs

L$ - ligne à faire défiler

Commentaires

4Ø : Le texte est complété d'un espace pour effacer le dernier caractère de la dernière ligne affichée.

8Ø : Affichage à partir de la première colonne de l'écran de la partie de la ligne qui dépasserait à droite, puis à la colonne I, du début de la ligne jusqu'à la dernière colonne.

9Ø : Boucle d'attente.

EXERCICE 3.8

Programme

```
10 CLS
20 PRINT:PRINT "TAPER UNE PHRASE A LA FORME NEGATIVE
:"
30 LINE INPUT A$
40 REM
50 REM DEBUT DE LA BOUCLE
60 FOR I=LEN(A$) TO 1 STEP -1
70 IF MID$(A$,I,4)=" NE " THEN A$=LEFT$(A$,I)+MID$(A$
,I+4)
```

```
80 IF MID$(A$,I,3)=" N'" THEN A$=LEFT$(A$,I)+MID$(A$,
I+3)
90 IF MID$(A$,I,5)=" PAS " THEN A$=LEFT$(A$,I)+MID$(A
$,I+5)
100 NEXT I
110 IF RIGHT$(A$,4)=" PAS" THEN A$=LEFT$(A$,LEN(A$)-4
)
120 IF LEFT$(A$,3)="NE " THEN A$=MID$(A$,4)
130 IF LEFT$(A$,2)="N'" THEN A$=MID$(A$,3)
140 PRINT
150 PRINT "LA PHRASE A LA FORME AFFIRMATIVE EST :"
160 PRINT A$
170 PRINT
180 END
```

Variables

I - compteur

A$ - variable

Commentaires

60-100 : L'un des mots NE, N' ou PAS est comparé à la partie de A$ de la même longueur : si une correspondance est trouvée, alors le mot est retiré.

110 : Test du cas où PAS se trouverait en fin de phrase.

120-130 : Test du cas où NE ou N' se trouvent en début de phrase.

Remarque : LEFT$(A$,I) donne les premiers caractères de A$ jusqu'au Ième inclus, ce qui dans le programme permet de laisser un espace entre les deux mots entourant celui à éliminer.

EXERCICE 3.9

Programme

```
10 CLS
20 INPUT "CHAINE DE CARACTERES ";C$
30 PRINT
40 INPUT "CARACTERES A INSERER ";I$
50 INPUT "POSITION D'INSERTION ";P
60 IF P<0 OR P>LEN(C$)+1 THEN 50
70 C$=LEFT$(C$,P-1)+I$+MID$(C$,P)
```

```
80 PRINT
90 PRINT "RESULTAT    "
100 PRINT C$
110 PRINT
120 END
```

Variables

P - position d'insertion

C$ - chaîne de caractères initiale

I$ - chaîne à insérer

Commentaires

60 : Teste la validité de la position d'insertion

70 : Insertion

EXERCICE 3.10

Programme

```
10 CLS:CLEAR 500
20 REM
30 REM ENTREES
40 PRINT:PRINT "TAPER LE TEXTE (MAX:128 CARACTERES) "
50 PRINT:LINE INPUT L$
60 IF L$="" OR LEN(L$)>128 THEN 40
70 PRINT
80 LINE INPUT "MOT A REMPLACER : ";R$
90 LINE INPUT "MOT A METTRE A SA PLACE : ";M$
100 IF LEN(R$)>128 OR LEN(M$)>128 THEN 60
110 REM
120 REM DEBUT DE LA RECHERCHE
130 I=1:F=0:R=LEN(R$):M=LEN(M$)
140 REM
150 REM DEBUT DE LA BOUCLE DE RECHERCHE
160 IF MID$(L$,I,R)<>R$ THEN 260
170 F=1
180 PRINT:IF LEN(L$)-R+M>128 THEN PRINT "REMPLACEMENT
 IMPOSSIBLE":GOTO 310
190 REM
200 REM DOIT-ON REMPLACER?
210 PRINT:PRINT LEFT$(L$,I-1);" ***";R$;"*** ";MID$(L
$,I+R)
220 PRINT:PRINT "VOULEZ-VOUS REMPLACER CELUI-LA(O/N)
?";
230 A$=INPUT$(1):IF A$<>"N" AND A$<>"O" THEN 230
```

```
235 PRINT A$
240 IF A$="N" THEN 260
250 L$=LEFT$(L$,I-1)+M$+MID$(L$,I+R):I=I+M
260 I=I+1:IF I<=LEN(L$)-R+1 THEN 150
270 REM
280 REM AFFICHAGE DU RESULTAT
290 PRINT
300 IF F=0 THEN PRINT "LE MOT N'EST PAS DANS LA LIGNE
."
310 PRINT:PRINT L$
320 END
```

Variables

I - compteur

F - flag (drapeau) : Ø si le mot à remplacer n'est pas dans la ligne

R - longueur de R$

M - longueur de M$

A$ - touche frappée au clavier

L$ - texte

R$ - mot à remplacer

M$ - mot à substituer

Commentaires

6Ø : Si la ligne ne contient aucun mot le traitement de la ligne est impossible.

13Ø : Initialisation du compteur, du drapeau et des longueurs des mots.

16Ø : Le mot à remplacer se trouve-t-il à la position I ? Si non, le compteur est incrémenté à la ligne 26Ø.

17Ø : Si oui, le drapeau est levé indiquant que le mot existe dans la phrase (F est mis à 1).

21Ø : Affichage de la ligne avec le mot à remplacer encadré d'étoiles.

3ØØ : Si F est égal à Ø, cela indique que le mot à remplacer n'a pas été trouvé (la ligne 17Ø n'a pas été exécutée).

EXERCICE 3.11

Programme

```
10 DIM E(40)
20 REM
25 ESP$="
30 REM ENTREE DE LA LIGNE
40 CLS
50 PRINT "TAPEZ LA LIGNE (MAX:40 CAR.) :"
60 LINE INPUT P$
65 IF LEFT$(P$,1)=" " THEN P$=MID$(P$,2):GOTO 65
70 IF LEN(P$)>40 THEN 50
80 REM
90 REM RECHERCHE DES ESPACES
100 N=0
105 FLG=0
110 I=1
120 IF MID$(P$,I,1)<>" " THEN FLG=0 ELSEIF FLG=0 THEN
 N=N+1:E(N)=I:FLG=1 ELSE P$=LEFT$(P$,I-1)+MID$(P$,I+1
):I=I-1
125 I=I+1
130 IF I<=LEN(P$) THEN 120
140 IF N=0 THEN PRINT "PAS ASSEZ DE MOTS. ":GOTO 240
150 REM JUSTIFICATION
160 B=40-LEN(P$)
170 FOR I=N TO 1 STEP -1
180 P$=LEFT$(P$,E(I))+LEFT$(ESP$,B@I)+MID$(P$,E(I)+1)
190 B=B-B@I
200 NEXT I
210 REM
220 REM AFFICHAGE
230 PRINT:PRINT P$
240 END
```

Variables

I - compteur

N - nombre d'espaces dans P$ (nombre d'éléments de E (40))

B - nombre d'espaces à ajouter à P$ pour faire la justification

P$ - ligne à justifier

E(40) - tableau des positions des espaces contenus initialement dans P$.

Commentaires

65 : Suppression des espaces éventuels en début de ligne.

90-130 : Les positions des espaces de P$ sont mises, par ordre croissant, dans E(40) et les espaces comprenant plusieurs blancs sont comprimés à un seul blanc.

140 : Si P$ ne contient aucun espace, il est impossible de justifier la ligne.

160 : Le nombre d'espaces à ajouter à P$ est mis dans B.

170-200: Les espaces sont ajoutés à partir de la fin de P$, pour que les éléments précédents de E(40) contiennent les positions réelles des espaces correspondants.

EXERCICE 3.12

Programme

```
10 CLS
20 PRINT "ENTREZ LA LIGNE DE TEXTE "
30 PRINT:LINE INPUT L$
40 IF L$="" THEN 20
50 REM
60 REM CREATION DE LA CHAINE INVERSE
70 I$=""
80 FOR I=LEN(L$) TO 1 STEP -1
90 I$=I$+MID$(L$,I,1)
100 NEXT I
110 REM
120 REM CREATION DE LA CHAINE ALEATOIRE
130 T$=L$:A$=""
140 FOR I=1 TO LEN(L$)
150 A=RND*LEN(T$)+.5
160 A$=A$+MID$(T$,A,1)
170 T$=LEFT$(T$,A-1)+MID$(T$,A+1)
180 NEXT I
190 REM
200 REM AFFICHAGE
210 PRINT:PRINT
220 PRINT L$:PRINT I$ :PRINT A$
230 PRINT
240 END
```

Variables

I - compteur

A - position aléatoire d'un caractère dans T$

L$ - ligne de texte

I$ - ligne inversée

A$ - ligne aléatoire

T$ - variable

Commentaires

130 : La ligne de texte est recopiée dans T$ pour pouvoir être manipulée sans perdre l'original. A$ est initialisé.

150 : RND (1) * LEN (T$) + 0.5 donne un nombre aléatoire entre 0.5 et la longueur de T$ augmentée de 0.5.

160-170 : Le Aième caractère de T$ est ajouté à A$ puis supprimé de T$.

PETIT EXERCICE 3.14

Programme

```
10 CLS
20 FOR I=32 TO 191
30 PRINT CHR$(I);
40 NEXT I
50 END
```

Variable

I - compteur (code ASCII du caractère à afficher)

Commentaire

30 : Les caractères s'affichant sur l'écran sont ceux dont le code ASCII est compris entre 32 et 191.

EXERCICE 3.15

Programme

```
10 CLS
15 CLEAR 500
20 LINE INPUT "ENTREZ UNE PHRASE :";L$
30 REM
40 REM PASSAGE EN MINUSCULE
50 REM
60 V$=""
70 FOR I=1 TO LEN(L$)
80 A$=MID$(L$,I,1):A=ASC(A$)
90 IF A>64 AND A<91 THEN A=A+32
100 V$=V$+CHR$(A)
110 NEXT I
120 PRINT:PRINT
130 PRINT L$
140 PRINT V$
150 END
```

Variables

I - compteur

A - code ASCII du caractère à tester

L$ - phrase à transformer

V$ - phrase transformée

Commentaire

90 : Si A$ est une majuscule, son code ASCII est augmenté de 32.

EXERCICE 3.16

Programme

```
10 DIM T(50)
20 REM
30 REM ENTREE DES DONNEES
40 INPUT "COMBIEN DE VALEURS (1 A 50) ";N
50 IF N<1 OR N>50 THEN 30
60 FOR I= 1 TO N :PRINT "VALEUR ";I;
70 INPUT T(I):NEXT I
80 REM
90 REM CALCUL DU POURCENTAGE
100 S=0
110 FOR I=1 TO N:S=S+T(I):NEXT I
```

```
120 FOR I=1 TO N:T(I)=T(I)*100/S:NEXT I
130 REM
140 REM AFFICHAGE
150 CLS
160 FOR I=1 TO N
170 PRINT "VALEUR ";I;TAB(13);":";
180 PRINT USING "###.##";T(I)
190 NEXT I
200 END
```

Variables

I - compteur

N - nombre de valeurs

S - somme des valeurs

T(5Ø) - tableau des valeurs

Commentaire

Pour calculer le pourcentage, il faut calculer la somme des valeurs (ligne 1ØØ-11Ø) puis diviser chaque valeur par cette somme et multiplier par 1ØØ (ligne 12Ø). Les pourcentages sont arrondis et alignés sur le point décimal à l'affichage en utilisant l'instruction PRINT USING (ligne 17Ø-18Ø).

EXERCICE 3.17

Programme

```
10 CLEAR 500
20 DIM L(26),T$(10)
30 REM
40 REM ENTREE DU TEXTE
50 CLS
60 INPUT "NOMBRE DE PHRASES (1 A 10) ";N
70 IF N<1 OR N>10 THEN 60
80 FOR I=1 TO N:PRINT "PHRASE ";I;":"
90 LINE INPUT T$(I):NEXT I
100 CLS
110 REM
120 REM CALCUL
130 FOR J=1 TO N
140 FOR K=1 TO LEN(T$(J))
150 LSC=ASC(MID$(T$(J),K,1))
160 IF LSC>64 AND LSC<91 THEN L(LSC-64)=L(LSC-64)+1
170 NEXT K
180 NEXT J
```

```
190 REM
200 S=0:FOR I=1 TO 26:S=S+L(I):NEXT I
210 FOR I=1 TO 26:L(I)=INT(L(I)*10000/S+.5)/100:NEXT
I
220 REM
230 REM AFFICHAGE
240 PRINT
250 FOR I=1 TO 13
260 PRINT CHR$(64+I);" :";
270 PRINT USING "###.##";L(I);
280 PRINT TAB(20);CHR$(77+I);" :";
290 PRINT USING "###.##";L(I+13)
300 NEXT I
310 END
```

Variables

I,J,K - compteurs

N - nombre de phrases

S - nombre total de lettres dans T$ (10)

LSC - code ASCII du caractère testé

L(26) - tableau des pourcentages d'apparition des lettres de l'alphabet

T$(10) - tableau contenant les phrases du texte

Commentaires

160 : Si le caractère testé est une lettre (majuscule), alors la valeur de l'élément dont l'indice est égal à LSC-64 est incrémenté de 1. Par exemple : si la lettre testée est H (code ASCII 72), l'élément d'indice 8 du tableau L est incrémenté de 1.

200 : Calcul du nombre total de lettres de l'alphabet contenues dans T$(10).

210 : Calcul du pourcentage arrondi à 2 décimales.

250-300 : Affichage sur deux colonnes des lettres de l'alphabet suivies de leur pourcentage de présence ajusté sur le point décimal.

EXERCICE 3.18

Programme

```
10 CLS
20 INPUT "JOUR (DE 1 A 31) ";J
30 INPUT "NUMERO DU MOIS (DE 1 A 12) ";M
40 INPUT "ANNEE ";A
50 IF J<1 OR J>31 OR M<1 OR M>12 THEN 20
60 FOR I=1 TO M:READ M$:NEXT I
70 D$=STR$(J)+" "+M$+STR$(A)
80 PRINT:PRINT D$:PRINT
90 END
200 DATA JANVIER,FEVRIER,MARS,AVRIL,MAI,JUIN,JUILLET
210 DATA AOUT,SEPTEMBRE,OCTOBRE,NOVEMBRE,DECEMBRE
```

Variables

I - compteur

J - numéro du jour

M - numéro du mois

A - année

D$ - date

M$ - le Mième mois

Commentaires

50 : Vérification de la validité des entrées

60 : Recherche du Mième mois dans les lignes de DATA

70 : Création de la date

EXERCICE 3.19

Programme

```
10 CLEAR 400
20 CLS
25 ME$="ERREUR (!!)"
30 INPUT "DATE (EX: 11 JANVIER 1963)";D$
40 REM
50 REM RECHERCHE DU JOUR DU MOIS ET DE L'ANNEE
60 FOR I=1 TO LEN(D$):IF MID$(D$,I,1)=" " THEN 100
70 NEXT I
80 PRINT ME$:GOTO 30
```

```
100 J=VAL(LEFT$(D$,I-1)):IF J>31 OR STR$(J)<>" "+LEFT
$(D$,I-1) THEN PRINT ME$:GOTO 30
110 D$=MID$(D$,I+1)
120 FOR I=1 TO LEN(D$):IF MID$(D$,I,1)=" " THEN 130
125 NEXT I
130 A=VAL(MID$(D$,I+1)):IF STR$(A)<>" "+MID$(D$,I+1)
THEN PRINT ME$:GOTO 30
140 D$=LEFT$(D$,I-1)
150 REM
160 REM RECHERCHE DU NUMERO DU MOIS
170 RESTORE
180 FOR M=1 TO 12:READ A$
190 IF A$=D$ THEN 210
195 NEXT M
200 PRINT ME$:GOTO 30
210 REM
215 REM AFFICHAGE DU RESULTAT
220 PRINT
230 PRINT J;"/";M;"/";A
240 PRINT
250 END
500 DATA JANVIER,FEVRIER,MARS,AVRIL,MAI,JUIN,JUILLET,
AOUT
510 DATA SEPTEMBRE,OCTOBRE,NOVEMBRE,DECEMBRE
```

Variables

I - compteur

J - numéro du jour

M - numéro du mois

A - année

D$ - date

A$ - variable

ME$ - message d'erreur

Commentaires

60 : Recherche du premier espace dans D$.

100 : Test de la validité du numéro du jour.

120 : Recherche du deuxième espace dans D$.

130 : Test de la validité du numéro de l'année.

180-200 : Recherche du numéro du mois

EXERCICE 3.20

Programme

```
10 DIM C$(9,5)
20 REM
30 REM LECTURE DES CHIFFRES
40 FOR I=0 TO 9
50 FOR J=0 TO 5
60 READ C$(I,J)
70 NEXT J,I
80 REM
90 REM ENTREE DES DONNEES A AFFICHER
100 CLS
110 INPUT "VOTRE NOMBRE (ENTRE 0 ET 99999) ";M$
120 M=LEN(M$)
130 IF M<1 OR M>5 THEN 110
140 REM AFFICHAGE
150 FOR I=0 TO 5
160 FOR J=1 TO M
170 D=VAL(MID$(M$,J,1))
180 PRINT C$(D,I);"  ";
190 NEXT J
200 PRINT
210 NEXT I
220 END
500 DATA " *** ","*   *","*   *","*   *","*   *"," **
* "
505 REM
510 DATA " **  ","  *  ","  *  ","  *  ","  *  ","***
**"
515 REM
520 DATA " *** ","*   *","   * ","  *  "," *   ","***
**"
525 REM
530 DATA " *** ","*   *","  ** ","    *","*   *"," **
* "
535 REM
540 DATA "    *","   * ","  *  "," * * ","*****","
* "
545 REM
550 DATA "**** ","*    ","**** ","    *","    *","***
* "
555 REM
560 DATA " *** ","*    ","**** ","*   *","*   *"," **
* "
```

```
565 REM
570 DATA "*****","*   *","   * ","  *  "," *   "," *   "
575 REM
580 DATA " *** ","*   *"," *** ","*   *","*   *"," *** "
585 REM
590 DATA " *** ","*   *"," ****","    *","    *"," *** "
```

Variables

I,J - compteurs

D - chiffre à afficher

M - nombre à afficher

M$ - nombre à afficher avec les nouveaux graphismes

C$(9,5) - tableau contenant les dessins des chiffres (le premier indice (Ø-9) désigne le chiffre et le second (Ø-5) une ligne du dessin.

Commentaires

3Ø-7Ø : Lecture des dessins (la ligne 5ØØ contient le dessin du chiffre Ø, la ligne 51Ø celle du cniffre 1, etc.).

15Ø : I repère les 6 lignes (0-5) des graphismes

18Ø : Affichage de la ième ligne correspondant au jème chiffre suivie de deux espaces pour séparer les différents chiffres.

19Ø : Permet le passage à la ligne suivante après M affichages.

EXERCICE 4.2

Programme

```
10 CLS
20 INPUT "DONNEZ UN NOMBRE (ENTRE 0 ET 24) ";L
30 IF L<0 OR L>24 THEN 20
40 INPUT "DONNEZ UN NOMBRE (ENTRE 0 ET 39) ";C
50 IF C<0 OR C>39 THEN 40
60 CLS
70 LOCATE 0,0,0
80 LINE (0,L)-(39,L) "*"
90 LINE (C,0)-(C,24) "*"
100 IF INKEY$="" THEN 100
110 END
```

Variables

L,C - variables

Commentaires

70 : Suppression de l'affichage du curseur clignotant.

80 : Affichage de la ligne d'étoiles à la ligne L.

90 : Affichage de la colonne d'étoiles à la colonne C.

100 : Pour éviter une remontée des lignes et l'effacement partiel de l'écran, provoqué par l'apparition de OK, une touche est attendue au clavier pour arrêter le programme. Puisque la touche frappée n'importe pas, elle n'est pas conservée dans une variable.

EXERCICE 4.3

Programme

```
10 DIM N(15)
20 CLS
30 REM
40 REM ENTREE DES NOMBRES
50 FOR I=1 TO 15
60 PRINT "NOMBRE ";I;" (0-100)";
70 INPUT N(I)
80 IF N(I)<0 OR N(I)>100 THEN 60
90 NEXT I
100 REM
110 REM AFFICHAGE DE L'HISTOGRAMME
120 CLS:LOCATE 0,0,0
130 FOR I=1 TO 30
140 C=((I/2) MOD 2)+1
150 IF N(I/2)<=2.5 THEN 170
160 LINE (I,21)-(I,22-20*N(I/2)/100) "*",C
170 NEXT I
190 IF INKEY$="" THEN 190
200 COLOR 7,0
210 END
```

Variables

C - couleur

I - compteur

N(15) - tableau des valeurs

Commentaires

130 : La variation de I a été choisie égale à 30 pour permettre d'afficher des bandes de largeur double.

140 : Changement de couleur pour chaque nouvelle valeur de N(I) (une bande rouge, une bande jaune).

150 : Si N(I/2) est inférieur à 2,5 on affiche une bande de hauteur nulle.

160 : La bande est affichée sur une hauteur proportionnelle à la valeur N(I/2) à partir de la position 21. Cette hauteur est exacte lorsque l'expression 20*N(I/2)/100 est entière. Sinon, l'instruction LINE arrondit à l'entier le plus proche.

EXERCICE 4.4

Programme

```
10 DIM N(24)
20 CLS
30 INPUT "POUR COMBIEN DE NOMBRES (1-24)";N
40 IF N<1 OR N>24 THEN 30
50 FOR I=1 TO N
60 PRINT "NOMBRE ";I;" (POSITIF) ";
70 INPUT N(I)
80 IF N(I)<0 THEN 60
90 NEXT I
100 REM
110 REM RECHERCHE DE LA PLUS GRANDE VALEUR ENTREE
120 G=N(1)
130 FOR I=2 TO N
140 IF G<N(I) THEN G=N(I)
150 NEXT I
160 REM
170 REM CALCUL DU PAS
180 VMAX=INT(G)+1
190 PAS=VMAX/20
200 REM
210 REM AFFICHAGE DE L'HISTOGRAMME
220 CLS:LOCATE 0,0,0:C=1
225 PX=12-N/2
230 FOR I=1 TO N
240 IF N(I)<PAS/2 THEN 260
250 LINE (PX+I,21)-(PX+I,22-N(I)/PAS)"*",C+2
260 C=-C         'C PASSE ALTERNATIVEMENT DE -1 A +1
```

```
270 NEXT I
280 FOR I=1 TO N
290 LOCATE 27,I-1:PRINT "N(";MID$(STR$(I),2);") =";N(
I)
300 NEXT I
310 IF INKEY$="" THEN 310
320 COLOR 7,0
330 END
```

Variables

C	- couleur
I,J	- compteur
N	- nombre de valeurs dans le tableau N(24)
G	- plus grande valeur du tableau N(24)
PX	- position de la première colonne de l'histogramme
PAS	- valeur correspondant à une étoile sur l'histogramme
VMAX	- la valeur arrondie de G par excès
N(24)	- tableau des valeurs

Commentaires

190 : La valeur correspondant à une étoile sur l'histogramme est égale à la valeur maximum arrondie, divisée par le nombre d'étoiles maximum.

225 : Calcul de la position de la première colonne de l'histogramme pour permettre son cadrage horizontal sur l'écran.

240 : Si la valeur à représenter est comprise entre 0 et PAS/2, elle est considérée comme égale à 0.

250 : Affichage de la colonne d'étoiles correspondant à la Ième valeur.

260 : Ecriture de l'instruction REM sous forme abrégée à la suite d'une instruction.

290 : Affichage des valeurs numériques représentées sur l'histogramme.

310 : Attend une touche avant de s'arrêter car l'arrêt du programme va provoquer une remontée des lignes.

EXERCICE 4.5

Programme

```
10 DIM N(1,9)
20 CLS
30 REM
40 REM ENTREE DES DONNEES
50 FOR I=0 TO 9
60 PRINT "VALEURS POUR"+STR$(I+1971)+" (PREMIERE,SECO
NDE) :";
70 INPUT N(0,I),N(1,I)
80 IF N(0,I)<0 OR N(1,I)<0 THEN 60
90 NEXT I
100 REM
110 REM RECHERCHE DE LA PLUS GRANDE VALEUR
120 G=0
130 FOR I=0 TO 9
140 IF N(0,I)>G THEN G=N(0,I)
150 IF N(1,I)>G THEN G=N(1,I)
160 NEXT I
170 IF G=0 THEN END
180 VMAX=INT(G)+1:PAS=VMAX/20
190 REM AFFICHAGE DE L'HISTOGRAMME
200 CLS
210 FOR I=0 TO 9
215 IF N(0,I)<PAS/2 THEN 225
220 LINE (3*I+3,20)-(3*I+3,21-20*N(0,I)/G) " ",0,1
225 IF N(1,I)<PAS/2 THEN 240
230 LINE (3*I+4,20)-(3*I+4,21-20*N(1,I)/G) " ",0,3
240 COLOR 7,0:LOCATE 3*I+2,21,0:PRINT I+71
250 NEXT I
260 COLOR 7,0
270 IF INKEY$="" THEN 270
280 END
```

Variables

I,J - compteurs

G - plus grande valeur du tableau N(1,9)

VMAX - valeur arrondie de G par excès

PAS - valeur correspondant à un pavé

N(1,9) - tableau des valeurs (N(Ø,x) - première série de dix valeurs, N(1,x) - deuxième série de dix valeurs).

Commentaires

50-90 : Entrée des deux séries de valeurs, année par année.

220-230 : Affichage d'une colonne de hauteur proportionnelle à la valeur à représenter en utilisant l'affichage d'espaces en couleur inverse.

EXERCICE 4.6

Programme

```
10 DIM V(36),N(36)
20 CLS:LOCATE 0,10:INPUT "ENTREZ UN NOMBRE ";R
30 R=RND(-R)
40 PRINT:INPUT "COMBIEN DE DES(1-6) ";N
50 IF N<1 OR N>6 THEN 40
60 CLS:LOCATE 0,0,0
70 REM
80 REM LANCER DES DES
85 PRINT "SOMME DES FACES SORTIES ":PRINT "POUR ARRET
ER, FRAPPER UNE TOUCHE.":PRINT
90 FOR I=1 TO 600
100 IF INKEY$<>"" THEN 175
110 S=0
115 FOR J=1 TO N
120 S=S+INT(6*RND)+1
130 NEXT J
140 V(S)=V(S)+1
142 PRINT S;
145 G=0
150 FOR J=N TO 6*N
155 N(J)=V(J)/I
160 IF G<N(J) THEN G=N(J)
165 NEXT J
170 NEXT I
175 CLS
180 FOR J=N TO 6*N
185 PAS=10*(J-N)+8
190 LINE (PAS,160)-(PAS,163)
195 FOR K=PAS TO PAS+10
200 LINE (K,160)-(K,160-150*N(J)/G)
205 NEXT K
210 NEXT J
220 PAS=PAS+10
```

```
230 LINE (PAS,160)-(PAS,163)
240 LOCATE 0,21:PRINT N
245 PRINT
250 PRINT TAB(2*N-2)"VALEUR MAX. = ";G
270 IF INKEY$="" THEN 270
280 END
```

Variables

R - Variable d'initialisation

I,J,K - compteurs

N - nombre de dés à lancer

S - somme des dés pour un lancer

PAS - position horizontale de la bande affichée

G - valeur maximale du tableau N(J)

V(36) - le tableau des sommes (V(s) contient le nombre de fois où la somme des dés a été égale à s).

N(36) - tableau des valeurs obtenues en divisant V(s) par le nombre total de tirages.

Commentaires

30 : Initialisation de la séquence des nombres fournis par l'instruction RND.

90-170 : Boucle de calcul et d'affichage des sommes pour un nombre maximum de 600 lancers.

100 : Possibilité de sortie de la boucle avant la fin des 600 lancers par pression d'une touche quelconque.

110-130 : Calcul de la somme des faces sorties.

150-165 : Pour obtenir un histogramme restant dans les limites de l'écran, on divise toutes les valeurs par le nombre total de lancers.

180-210 : Affichage de l'histogramme

190 : Affichage des graduations horizontales.

195-205 : Affichage des bandes de l'histogramme.

230 : Affichage de la dernière graduation.

240 : Affichage de la somme minimale de faces sorties (c'est-à-dire du nombre de dés).

EXERCICE 4.7

Programme

```
10 REM ENTREE DE L'ESPACEMENT
20 CLS
30 INPUT "LARGEUR DE LA MAILLE (1-198) ";E
40 IF E<1 OR E>198 THEN 30
50 CLS:LOCATE 0,0,0
55 N=INT(199/E)
60 REM
70 REM DROITES HORIZONTALES
80 REM DROITES VERTICALES
90 REM
100 FOR I=0 TO N
110 LINE (0,I*E)-(N*E,I*E)
120 LINE (I*E,0)-(I*E,N*E)
130 NEXT I
140 IF INKEY$="" THEN 140
150 END
```

Variables

I - compteur

N - nombre de mailles sur une ligne ou une colonne

E - largeur de la maille

Commentaires

100-130 : Affichage du quadrillage

EXERCICE 4.8

Programme

```
10 CLS:LOCATE 0,0,0
20 X=160:Y=100:F=3
30 REM
40 REM AFFICHAGE
50 PSET (X,Y) ,F
60 REM
70 REM ENTREE D'UNE TOUCHE
80 A$=INKEY$:IF A$="" THEN 80
90 AS=ASC(A$)
100 IF AS=8 THEN IF X<1 THEN X=0:GOTO 50 ELSE X=X-1:G
OTO 50
110 IF AS=9 THEN IF X>318 THEN X=319:GOTO 50 ELSE X=X
+1:GOTO 50
120 IF AS=10 THEN IF Y>198 THEN Y=199:GOTO 50 ELSE Y=
Y+1:GOTO 50
```

```
130 IF AS=11 THEN IF Y<1 THEN Y=0:GOTO 50 ELSE Y=Y-1:
GOTO 50
140 IF AS=28 THEN F=3:GOTO 80
150 IF AS=29 THEN F=0:GOTO 80
160 IF AS=13 THEN 10
170 END
```

Variables

F - flag (drapeau) : 3 si le point est éclairé (en jaune), Ø si le point est éteint.

X,Y - coordonnées du point à afficher

A$ - touche tapée par l'utilisateur

AS - code ASCII de la touche tapée.

Commentaires

2Ø : Initialisation au milieu de l'écran, avec affichage, (F=3)

1ØØ-13Ø : Les mouvements correspondants ne sont possibles que si le point n'est pas sur le bord de l'écran

14Ø : Une pression de la touche INS éclaire le point en jaune.

15Ø : Une pression de la touche EFF éteint le point.

16Ø : Une pression de la touche ENTREE efface l'écran et repositionne le point au centre de l'écran.

17Ø : Une pression sur une touche quelconque autre que celles testées arrête le jeu.

EXERCICE 4.9

Programme

```
10 CLS:LOCATE 0,0,0
15 PI=3.1415926535
20 REM AXES
30 LINE (20,15)-(20,185)
40 LINE (15,100)-(295,100)
50 PSET (20,20)
60 FOR I=20 TO 290
70 X=4*PI*(I-20)/360
80 Y=100-80*SIN(X)
```

```
90 LINE -(I,Y)
100 NEXT I
110 REM
120 REM GRADUATIONS
130 I=1
140 K=(90*I/PI)+20
150 IF K>290 THEN 180
160 LINE (K,100)-(K,97)
170 I=I+1:GOTO 140
180 REM
190 LINE (20,20)-(23,20)
200 LINE (20,180)-(23,180)
210 LOCATE 1,13:PRINT "0"
220 LOCATE 6,13:PRINT"1"
230 LOCATE 1,2:PRINT "1"
240 IF INKEY$="" THEN 240
250 END
```

Variables

X,Y - coordonnées du point à afficher

I - compteur des graduations

PI - constante

K - position d'une graduation

Commentaires

30 : Tracé de l'axe vertical

40 : Tracé de l'axe horizontal

60-100 : Tracé de la courbe

160 : Tracé d'une graduation horizontale.

190-200 : Tracé des graduations verticales

210-230 : Affichage de l'origine et des unités sur chaque axe.

PETIT EXERCICE 4.10

Programme

```
10 CLEAR ,,1
20 CLS
30 LOCATE 0,0,0
40 DEFGR$(0)=24,24,24,24,24,126,60,24
50 ATTRB 1,1:LOCATE 0,22
60 FOR I=1 TO 6
```

```
65 C=(I MOD 2)+1
70 COLOR 0,C:PRINT I;
80 NEXT I
90 ATTRB 0,0:COLOR 7,0
100 ALEA=ALEA+1:IF INKEY$="" THEN 100
110 GOSUB 200
115 COLOR 5,7
120 S=S+(PN @ 6)+1
125 NJ=NJ+1
130 ATTRB 1,1:LOCATE 2,1:PRINT NJ
135 LOCATE 30,1:PRINT S:ATTRB 0,0
140 COLOR 7,0:GOTO 100
200 REM CHUTE DE LA FLECHE
210 PA=36*RND(-ALEA)
220 FOR I=3 TO 20
230 PN=PA+6*RND-3:IF PN<0 THEN PN=0 ELSE IF PN>35 THE
N PN=35
240 LOCATE PA,I-1,0:PRINT " "
250 LOCATE PN,I,0:PRINT GR$(0)
260 PA=PN
270 NEXT I
280 PLAY "DO"
290 LOCATE PN,I-1,0:PRINT " "
300 RETURN
```

Variables

I : compteur

C : couleurs du fond de la cible

ALEA : nombre aléatoire

S : score

PA : position ancienne de la flèche

PN : position nouvelle de la flèche

NJ : nombre de coups joués.

Commentaires

10 : Réservation de la place pour un caractère graphique.

40 : Définition du caractère graphique "flèche"

50 : Passage en caractères de dimension double.

60-80 : Affichage de la cible.

90 : Passage en caractères de dimension normale.

100 : Création d'un nombre aléatoire variant en fonction du temps de réaction du joueur.

115-140 : Affichage du nombre de coups joués et du score.

200-300 : Sous-programme de chute de la flèche.

210 : Calcul de la position horizontale de départ.

230 : Calcul de la position nouvelle et test de cadrage dans la zone de jeu.

240-250 : L'effet de mouvement est obtenu par effacement et réaffichage successifs de la flèche.

280 : Emission d'une note au moment de l'impact de la flèche.

290 : Effacement de la dernière position.

EXERCICE 4.11

Programme

```
10 CLEAR,,1
15 DEFGR$(0)=129,66,36,24,24,36,66,129
20 REM ENTREE DU COTE
30 INPUT "UN NOMBRE (1-9) ";C
35 IF C<1 OR C>9 THEN 30
40 CLS:PRINT
45 FOR I=1 TO C
50 FOR J=1 TO C
60 N=N+1
70 IF N>9 THEN PRINT N;ELSE PRINT " ";N;
80 NEXTJ
90 PRINT:PRINT
100 NEXT I
110 FOR N=1 TO C^2
120 FOR D=2 TO C
130 IF (N MOD D)<>0 OR D=N THEN 160
135 COL=N MOD C:LIG=(N@C)
140 IF COL=0 THEN COL=C:LIG=LIG-1
145 IF N>9 THEN M$=GR$(0)+GR$(0) ELSE M$=" "+GR$(0)
150 LOCATE 4*COL-3,2*LIG+1,0:PRINT M$
155 GOTO 170
160 NEXT D
170 NEXT N
180 LOCATE 0,22,1:END
```

Variables

I,J - compteurs

C - nombre de lignes et de colonnes du carré

N - nombre à tester

D - diviseur

C - colonne d'affichage

LIG - ligne d'affichage

MØ - caractère d'effacement

Commentaires

45-1ØØ : Affichage du carré

11Ø-17Ø : Test des nombres

145 : Le caractère d'effacement comprend 1 ou 2 croix selon que le nombre est inférieur ou supérieur à 1Ø.

EXERCICE 5.1

Programme

```
10 CONSOLE0,24:SCREEN7,0,6:CLEAR,,1
15 DEFGR$(0)=0,60,127,254,252,254,127,60
20 CLS
30 PRINT "QUEL NIVEAU DE DIFFICULTE (ENTRE 0 ET 5) ";
35 A$=INKEY$:IF A$="" OR A$<"0" OR A$>"5" THEN RD=RND
(-RD):GOTO 35
40 N=VAL(A$):PRINT N
45 CLS
50 LINE (0,3)-(39,3)"-":LINE (0,23)-(39,23)"-":CONSOL
E 4,22
60 REM INITIALISATIONS
65 CLS:LOCATE 0,0,0
70 FOR I=1 TO 10+10*N
80 X=INT(30*RND+7.5):Y=INT(RND*18+4.5)
90 PSET (X,Y) CHR$(127),2
100 NEXT I
110 X=0:Y=INT(RND*18+4.5):DY=0
120 PSET (X,Y) GR$(0),1
130 IF INKEY$="" THEN 130
140 REM DEBUT DU JEU
150 A$=INKEY$:IF A$="" THEN 180 ELSE A=ASC(A$)
155 IF A=11 THEN DY=-1:GOTO 180
160 IF A=9 THEN DY=0:GOTO 180
165 IF A=10 THEN DY=1:GOTO 180
170 IF A$="X" THEN 600
180 IF (Y<=4 AND DY=-1) OR (Y>=22 AND DY=1) THEN DY=0
190 IF SCREEN(X+1,Y+DY)=127 THEN 500
200 PSET (X+1,Y+DY) GR$(0):PSET(X,Y)" "
```

```
210 X=X+1:Y=Y+DY
220 IF X<39 GOTO 150
230 PSET (X,Y) " "
240 N=N+1:NBC=NBC+1
250 PLAY "DO"
260 LOCATE 2,1:ATTRB1,1:COLOR7,0:PRINT NBC:ATTRB0,0
270 GOTO 60
500 REM PERDU !
505 PSET(X,Y)" "
520 FOR T=1 TO 2
530 LOCATE 16,1:PRINT "PERDU":FOR TT=1 TO 40:NEXT TT:
LINE(16,1)-(21,1)" ":FOR TT=1 TO 20:NEXT TT
540 PLAY "O4L8DODODODOSO":NEXT T
545 CONSOLE 4,22
550 IF N>=1 THEN N=N-1
560 FOR I=1 TO 500:NEXT I
570 GOTO 60
600 CONSOLE0,24:ATTRB0,0:COLOR7,0
610 END
```

Variables

A$: caractère correspondant à la touche frappée

RD : variable d'initialisation

N : niveau de difficulté

X,Y : coordonnées d'affichage

DY : déplacement vertical

NBC : score

I,T,TT : compteurs

Commentaires

15 : Création du caractère "glouton".

35 : Test de validité de la touche tapée et initialisation de l'instruction RND.

70-100 : Positionnement des obstacles sur la zone de jeu.

120 : Affichage du glouton à la position de départ.

130 : Attente de la pression d'une touche pour lancer le jeu.

150-170 : Analyse de la touche frappée.

180 : Si le glouton touche l'un des bords horizontaux, il poursuit sa course le long de ce bord.

190 : Test de la rencontre du glouton avec un obstacle.

2ØØ	:	Déplacement du glouton.
22Ø	:	Test d'arrivée au but.
24Ø	:	Le niveau de difficulté est augmenté à chaque traversée réussie.
52Ø-54Ø	:	Boucle d'affichage du message "PERDU".
55Ø	:	Le niveau de difficulté est diminué après chaque échec.

EXERCICE 5.2

Programme

```
10 SCREEN7,0,0:CLEAR,,2
15 DIM N$(1),J%(2,2),TABC(50),TABS(50)
20 CLS
25 DEFGR$(0)=24,66,66,129,129,66,66,24
30 DEFGR$(1)=24,36,66,129,129,66,36,24
35 PRINT "VOULEZ-VOUS JOUER AU MORPION (O/N) ";
40 RD=RD+1:IF INKEY$<>"O" THEN 40
45 LOCATE 0,10:INPUT "PRENOMS DES DEUX JOUEURS (10 CA
R. MAX.) ";N$(0),N$(1)
50 IF LEN(N$(0))>10 OR LEN(N$(1))>10 THEN 20
55 NBP=50:X1=48:Y1=16:LAR=96:L3=LAR/3:L6=LAR/6
60 LOCATE 0,15:PRINT "PATIENCE..JE PREPARE LA GRILLE
DE JEU !":GOSUB 1000
65 CLS:LOCATE 0,0,0
70 FOR I=0 TO 3
75 LINE (X1,Y1+L3*I)-(X1+LAR,Y1+L3*I)
80 LINE (X1+L3*I,Y1)-(X1+L3*I,Y1+LAR)
90 NEXT I
100 FOR I=0 TO 2:FOR J=0 TO 2
120 BOXF (X1+L3*J+10,Y1+L3*I+10)-(X1+L3*J+22,Y1+L3*I+
22),4
125 J%(I,J)=2
130 NEXT J,I
135 LOCATE 20,5,0:PRINT N$(0);SPC(1);" A LES ";GR$(0)
140 LOCATE 20,8,0:PRINT N$(1);SPC(1);" A LES ";GR$(1)
150 JO%=INT(RND(-RD)+.5)
160 LOCATE 6,20:PRINT N$(JO%);SPC(2);" JOUE";SPC(15):
PLAY "DO"
170 GOSUB 300 'JEU
175 IF TRIC=1 THEN TRIC=0:GOTO 160
180 J%(LI,CO)=JO%
190 GOSUB 800 'TEST
200 IF G<>1 AND NBCOUP=8 THEN 250
210 IF G=0 THEN JO%=-JO%+1:NBCOUP=NBCOUP+1:GOTO 160
220 LOCATE 3,20
```

```
230 PRINT N$(JO%);" A GAGNE...BRAVO!!":PLAY "DOREMIRE
DO"
240 END
250 LOCATE 3,20
260 PRINT SPC(8);"MATCH NUL ";SPC(10):PLAY "SISIDO"
270 END
300 REM SAISIE PAR CRAYON OPTIQUE
310 INPUTPEN X,Y
320 FOR I=0 TO 2:FOR J=0 TO 2
330 XINI=X1+L3*J+8:YINI=Y1+L3*I+8
340 IF X>XINI AND X<XINI+16 AND Y>YINI AND Y<YINI+16
THEN 500
350 NEXT J,I
360 TRIC=1
370 RETURN 'SP1
380 REM
500 LI=I
510 CO=J
520 IF J%(LI,CO)=2 THEN 550
530 TRIC=1
540 RETURN 'SP1
545 REM
550 IF JO%=0 THEN 700
560 XINI=X1+L6+CO*L3
570 YINI=Y1+LI*L3
600 LINE (XINI,YINI)-(XINI+L6,YINI+L6)
610 LINE -(XINI,YINI+2*L6)
620 LINE -(XINI-L6,YINI+L6)
630 LINE -(XINI,YINI)
640 RETURN 'SP1
650 REM
700 XINI=X1+CO*L3+L6
710 YINI=Y1+LI*L3+L6
720 PSET (XINI+13,YINI)
730 FOR RA=0 TO NBP
740 LINE -(XINI+TABC(RA),YINI+TABS(RA))
750 NEXT RA
760 RETURN 'SP1
800 REM
810 REM TEST DU COUP GAGNANT
820 IF (J%(0,2)=JO% AND J%(2,0)=JO% OR J%(0,0)=JO% AN
D J%(2,2)=JO%) AND J%(1,1)=JO% THEN G=1:RETURN 'SP2
830 REM
850 FOR J=0 TO 2
860 IF J%(LI,J)<>JO% THEN 900
870 NEXT J
880 G=1:RETURN 'SP2
```

```
890 REM
900 FOR I=0 TO 2
910 IF J%(I,CO)<>JO% THEN 950
920 NEXT I
930 G=1:RETURN 'SP2
940 REM
950 G=0:RETURN 'SP2
1000 REM
1010 FOR RA=0 TO NBP
1020 TABC(RA)=13*COS(RA*6.28/NBP)
1030 TABS(RA)=13*SIN(RA*6.28/NBP)
1040 NEXT RA
1050 RETURN
```

Variables

I,J	: compteurs
N$(1)	: tableau des noms des deux joueurs
J%(2,2)	: tableau des coups joués
TABC(50) et TABS(50)	: tableaux des coordonnées des points du cercle
RD	: variable d'initialisation
NBP	: nombre de points du cercle
X1,Y1,LAR,L3,L6	: paramètre de la grille de jeu
JO%	: joueur prêt à jouer
TRIC	: drapeau signalant que la case a déjà été jouée
LI	: numéro de ligne de la grille
CO	: numéro de colonne de la grille
XINI,YINI	: coordonnées
G	: drapeau signalant un coup gagnant
NBCOUP	: nombre de coups joués diminué de 1

Commentaires

70-90	: Dessin de la grille.
100-130	: Dessin des pavés de saisie du crayon optique.
125	: Initialisation du tableau J%
220-240	: Fin du programme avec joueur gagnant.
250-270	: Fin du programme par match nul.
300-370	: Sous-programme de saisie par crayon optique.

320-350 : Recherche de la case pointée par le crayon optique.

500

EXERCICE 5.3

Programme

```
10 REM LOTO
20 SCREEN 7,0
30 CLS
40 PRINT "   POUR OBTENIR UN TIRAGE TAPEZ SUR UNE TOU
CHE"
50 IF INKEY$="" THEN RD=RND:GOTO 50
60 COLOR 7,0:CLS
70 COLOR 1,3
75 LINE (3,0)-(15,0)" "
80 LINE (1,2)-(3,2)" "
85 FOR L=0 TO 9
90 FOR K=0 TO 4
95 IF K=0  THEN IF L=0 THEN 120 ELSE ABX=1 ELSE ABX=3
*K
100 LOCATE ABX,2*L+1,0:PRINT 10*K+L
110 LINE (ABX,2*L+2)-(ABX+3,2*L+2)" "
120 NEXT K,L
140 FOR I=1 TO 6
150 TIR(I)=INT(49*RND)+1
160 IF I=1 THEN 200
170 FOR J=1 TO I-1
180 IF TIR(I)=TIR(J) THEN 150
190 NEXT J
200 NEXT I
205 COLOR 4
210 FOR I=1 TO 6
220 IF TIR(I)<10 THEN ABX=1 ELSE ABX=3*(TIR(I) @ 10)
230 LOCATE ABX,2*(TIR(I) MOD10)+1:PRINT TIR(I);
240 NEXT I
250 COLOR 1
260 LOCATE 20,5
270 PRINT "AUTRE TIRAGE (O/N) ?";
280 A$=INKEY$:IF A$<>"O" AND A$<>"N" THEN RD=RND:GOTO
 280
290 IF A$="O" THEN 60
300 LOCATE 0,22:COLOR7,0:END
```

Variables

RD - variable d'initialisation
I,J,K,L - compteurs de boucle
ABX - numéro de colonne
A$ - touche tapée au clavier
TIR(6) - tableau des tirages dimensionné implicitement.

Commentaires

50 : Initialisation aléatoire de la fonction RND par exécutions successives de celle-ci.

85-120 : Affichage de la grille des nombres avec un seul espace entre chaque colonne de nombres.

140-200 : Tirage de six nombres.

170-190 : Comparaison du nombre tiré avec les I-1 premiers.

210-240 : Affichage avec une couleur différente des six nombres tirés.

EXERCICE 5.4

Programme

```
10 CLS:SCREEN 7,0,0
15 MESS1$="VOUS CALCULEZ VITE ET BIEN"
20 MESS2$="VOUS CALCULEZ VITE MAIS MAL. LE RESULTAT E
ST : "
25 MESS3$="VOUS N'ETES PAS ASSZ RAPIDE.LE RESULTAT ES
T: "
30 ATTRB 1,1:LOCATE 6,4
35 COLOR 1,3:PRINT "CALCUL MENTAL"
40 ATTRB 0,0:PRINT TAB(4)"VOUS AVEZ 10 SECONDES POUR
DONNER"SPC(3)
45 PRINT TAB(7)"LE RESULTAT DE L'OPERATION"SPC(7)
50 COLOR 7,0:CONSOLE 9,24
55 CLS:LOCATE 0,9
60 PRINT
65 PRINT "SI VOUS ETES PRET TAPEZ O ";:PLAY "DO"
80 X=RND:IF INKEY$<>"O" THEN 80
90 CLS
200 A1=INT(1000*RND)+1
210 A2=INT(1000*RND)+1
220 ON (10*RND)MOD4+1 GOSUB 1000,1100,1200,1300
```

```
240 GOSUB 2000
250 LINE (0,CSRLIN+1)-(39,CSRLIN+1)"-"
260 PRINT
270 INPUT "VOULEZ-VOUS CONTINUER (O/N)  ";REPCONT$
280 IF REPCONT$="O" THEN 50
290 IF REPCONT$<>"N" THEN LOCATE 0,CSRLIN-1:PRINT CHR
$(24);:PLAY "DO":GOTO 270
300 CONSOLE 0,24
310 END
1000 REM SOUS-PROGRAMME ADDITION
1010 OPE$=" + "
1020 RESO=A1+A2
1030 RETURN
1100 REM SOUS-PROGRAMME SOUSTRACTION
1110 OPE$=" - "
1120 RESO=A1-A2
1130 RETURN
1200 REM SOUS-PROGRAMME MULTIPLICATION
1210 OPE$=" * "
1220 RESO=A1*A2
1230 RETURN
1300 REM SOUS-PROGRAMME DIVISION
1310 OPE$=" / "
1320 RESO=A1/A2
1330 RETURN
2000 REM SOUS PROGRAMME DE TRAITEMENT
2020 PRINT TAB(10)A1;OPE$;A2;" = ";
2030 INPUTWAIT 2060;10,RES
2040 IF RES<>RESO THEN PRINT MESS2$   GOTO 2070
2045 PRINT MESS1$
2050 RETURN
2060 PRINT MESS3$
2070 COLOR ,,0
2080 LOCATE 10,CSRLIN:PRINT A1;OPE$;A2;" = "; RESO
2090 COLOR ,,0
2100 RETURN
```

Variables

MESS1$,MESS2$,MESS3$	- messages affichés à l'écran
X	- variable d'initialisation
A1,A2	- opérandes
REPCONT$	- réponse pour continuer ou arrêter le jeu
OPE$	- contient le symbole de l'opération
RESO	- résultat de l'opération

RES : réponse du joueur.

Commentaires

5Ø : L'instruction CONSOLE 9,24 définit la zone d'affichage active.

2Ø3Ø : L'instruction INPUTWAIT 2Ø6Ø;1Ø,RES affecte la réponse du joueur à la variable RES si celle-ci est donnée dans un délai de 10 secondes, sinon un branchement est effectué à la ligne 2Ø6Ø.

EXERCICE 5.5

Programme

```
10 REM
20 REM INITIALISATIONS
30 CLS:SCREEN7,0,0
40 CLEAR ,,3
50 DEFGR$(0)=129,66,36,24,24,36,66,129
60 DEFGR$(1)=24,60,126,255,255,126,60,24
70 DEFGR$(2)=24,36,36,66,66,255,0,0
100 REM
120 LOCATE 5,5:PRINT GR$(0)+GR$(0)+GR$(0);SPC(6);"+10
0 Points"
130 LOCATE 5,7:PRINT GR$(0)+GR$(1)+GR$(0);SPC(7);"+20
 Points"
140 LOCATE 5,9:PRINT GR$(0)+GR$(0)+GR$(1);SPC(8);"+2
Points"
150 LOCATE 5,11:PRINT GR$(1)+GR$(0)+GR$(0);SPC(8);"+2
 Points"
160 LOCATE 5,13:PRINT "AUTRE :   -20 Points"
180 LOCATE 5,15:PRINT "POUR COMMENCER TAPEZ UNE TOUCH
E";
190 RD=RND:IF INKEY$="" THEN 190
200 REM
210 CLS
220 ATTRB 1,1:COLOR1,2:LOCATE 9,2,0:PRINT "JACKPOT"
230 FOR I=1 TO 3
240 COLOR ,(I MOD2)+1:LOCATE 7+4*I,8:PRINT " "
250 NEXT I
255 COLOR ,2:LINE (14,12)-(17,12)" "
260 COLOR 6,0
270 BOX (9,5)-(22,17)"$"
360 FOR I=1 TO 3
370 V(I)=INT(3*RND)
380 NEXT I
```

```
400 REM
410 REM DEFILEMENT
415 FOR J=1 TO 3
420 FOR I=1 TO 30+V(J)
430 LOCATE 7+4*J,8:COLOR0,(J MOD2)+1:PRINT GR$(I MOD
3)
435 PLAY "L3A20DO"
440 NEXT I,J
450 A$="":FOR K=1 TO 3:A$=A$+RIGHT$(STR$(V(K)),1):NEX
T K
455 IF A$="000" THEN RES=100:GOTO 480
460 IF A$="010" THEN RES=20:GOTO 480
465 IF A$="100" OR A$="001" THEN RES=2:GOTO 480
470 RES=-20
480 SOMME=SOMME+RES
490 ATTRB 0,0:LOCATE 14,12:PRINT SPC(4-LEN(STR$(RES))
);STR$(RES)
500 LOCATE 25,10:PRINT SPC(5-LEN(STR$(SOMME)));STR$(S
OMME)
505 LOCATE 25,8:IF SOMME>0 THEN PRINT " GAIN" ELSE PR
INT "PERTE"
520 LOCATE 4,20,0:PRINT "VOULEZ-VOUS REJOUER (O/N)";
530 R$=INKEY$:IF R$<>"O" AND R$<>"N" THEN RD=RND:GOTO
 530
540 IF R$="N" THEN LOCATE 0,22:COLOR 7,0:END
550 LOCATE 4,20:PRINT SPC(26):ATTRB 1,1
560 GOTO 360
```

Variables

RD - variable d'initialisation

I,J - compteur de boucle

V(3) - tableau de trois valeurs tirées au hasard

RES - gain d'un coup

SOMME - gain d'une partie

A$ - chaîne de test du tirage

R$ - réponse pour arrêter ou continuer

Commentaires

50-70 : Définition des symboles qui s'affichent.

230-250 : Dessin des fenêtres de la machine.

255 : Dessin de la zone d'affichage du résultat d'un coup

270 : Dessin du cadre de la machine.

36Ø-38Ø : tirage de trois valeurs aléatoires.

42Ø : La valeur 3Ø+V(J) permet d'afficher le symbole correspondant à la valeur tirée après avoir fait "tourner" les trois symboles pendant au moins trente rotations.

49Ø-5Ø5 : Affichage du résultat d'un tirage et de la partie.

55Ø : Effacement de la question "voulez-vous rejouer (O/N) ?" par affichage d'espaces jusqu'à la colonne 26.

EXERCICE 5.6

Programme

```
10 CLEAR ,,2
30 DEFGR$(1)=0,126,126,126,126,126,126,0
40 CLS
50 SCREEN 7,0,0
70 I=0
80 I=I+1
90 READ A$:IF A$<>"FIN" THEN 80
100 NBMOT=I-1
130 RESTORE
140 REM
150 LOCATE 2,10,0
160 PRINT "POUR COMMENCER TAPEZ SUR UNE TOUCHE"
170 RD=RND:IF INKEY$="" THEN 170
180 REM
190 CLS
200 PRINT "VOUS AVEZ";INT(NBMOT/4);"SECONDES POUR LIR
E"
210 PRINT "CES";NBMOT;"MOTS"
220 LOCATE 0,5
230 FOR I=1 TO NBMOT
240 READ A$:PRINT A$,
250 NEXT I
260 RESTORE
270 FOR T=1 TO 100*NBMOT:NEXT T
280 CLS
290 LOCATE 0,10
300 INPUT "DONNEZ LE NOMBRE DE COUPS A JOUER";NBCOUP
310 NC=INT(RND*NBMOT)+1
320 FOR I=1 TO NC
330 READ MOT$
340 NEXT I
```

```
350 LMOT=LEN(MOT$)
360 DIM PCAR(LMOT)
370 CLS
380 ATTRB 1,1
390 COLOR 1,2:LOCATE 2,2:PRINT NBCOUP
400 COLOR 2,1:LOCATE 32,2:PRINT NBCOUP
410 COLOR 0,7
420 REM
430 FOR I=1 TO LMOT
440 LOCATE 2*I+15-LMOT/2,7
450 PRINT GR$(1)
460 NEXT I
470 ATTRB 0,0
480 COLOR 7,0
490 LOCATE 0,12
500 PRINT CHR$(24);
510 LINE INPUT "DONNEZ UN MOT    ";MOTPRO$
520 FOR I=1 TO LMOT
530 IF PCAR(I)=1 THEN 550
540 IF MID$(MOTPRO$,I,1)=MID$(MOT$,I,1) THEN GOSUB 10
00
550 NEXT I
560 IF NTROU=LMOT THEN 770
570 REM
580 NBCOUP=NBCOUP-1
590 ATTRB 1,1
600 COLOR 2,1:LOCATE 32,2
610 PRINT NBCOUP
630 ATTRB 0,0
640 COLOR 7,0
650 IF NBCOUP<>0 THEN 490
670 ATTRB 1,1
680 LOCATE 12,18
690 COLOR 4,7:PRINT "PERDU !!"
700 ATTRB 0,0
720 LOCATE 3,21
730 PRINT "LE MOT A DECOUVRIR ETAIT    ";MOT$
740 PLAY "DOREDO"
750 COLOR 7,0
760 END
770 ATTRB 1,1
780 LOCATE 12,18
790 COLOR 4,7:PRINT "GAGNE !!"
800 ATTRB 0,0
810 PLAY "DOREDO"
820 COLOR 7,0
830 END
```

```
1000 REM
1010 REM
1020 PCAR(I)=1
1030 NTROU=NTROU+1
1040 ATTRB 1,1
1050 LOCATE 2*I+15-LMOT/2,7
1060 COLOR 0,7
1070 PRINT MID$(MOT$,I,1)
1080 COLOR 7,0
1090 ATTRB 0,0
1100 RETURN
1500 DATA MAMAN,PAPA,PIERRE,CLOWN,BAROMETRE,ANALYSE
1510 DATA BALLON,ETAT,BAVARD,AZOTE,APPEL,VAISSEAU
1520 DATA GENERATEUR,ALMANACH
1530 DATA PLANETE,ROBOT,CINQ,VALEUR,PLACE,FINESSE,ICO
NE
1600 DATA FIN
```

Variables

I,T	-	compteurs
A$	-	mots cachés
NBMOT	-	nombre de mots de la liste
RD	-	variable d'initialisation
NBCOUP	-	nombre de coups joués.
NC	-	position du mot à trouver dans la liste
LMOT	-	longueur du mot à trouver
MOT$	-	le mot à trouver
MOTPRO$	-	le mot proposé par le joueur
PCAR	-	tableau des positions des caractères trouvés.
NTROU	-	nombre de lettres trouvées.

Commentaires

70-100	:	Calcul du nombre de mots de la liste.
230-250	:	Affichage des mots cachés.
270	:	Temporisation.
390-400	:	Affichage des nombres de coups.
430-460	:	Affichage de la grille à remplir.
500	:	Effacement de la ligne 12.
520-550	:	Comparaison, caractère par caractère, du mot proposé avec le mot à trouver.

530 : Le tableau PCAR sert à repérer les lettres déjà trouvées.

1000-1100: Sous-programme de traitement en cas de caractère trouvé : le Ième élément du tableau PCAR est mis à 1.

EXERCICE 5.7

Programme

```
10 CLEAR 500
20 DIM D$(200)
30 CLS
40 PRINT "1 - LECTURE DU FICHIER SUR CASSETTE"
50 PRINT "2 - AFFICHAGE SUR L'ECRAN"
60 PRINT "3 - INSERTION DE DONNEES"
70 PRINT "4 - SUPPRESSION DE DONNEES"
80 PRINT "5 - SAUVEGARDE DU FICHIER SUR CASSETTE"
90 PRINT
100 PRINT "0 - FIN"
110 PRINT:PRINT TAB(10)"CHOIX (0-5)";
115 INPUT R
120 IF R<=0 OR R>5 THEN END
130 ON R GOSUB 1000,1500,2000,2500,3000
140 GOTO 30
1000 REM
1010 REM LECTURE SUR CASSETTE
1020 REM
1030 INPUT "  LA  CASSETTE  EST-ELLE  POSITIONNEE  (
OUI/NON)";R$
1040 IF R$<>"OUI" THEN RETURN
1050 INPUT "NOM DU FICHIER D'ENTREE :";FII$
1070 OPEN "I",#1,FII$
1080 I=0
1090 I=I+1
1100 IF EOF(1)=-1 THEN 1140
1110 LINE INPUT #1,D$(I)
1120 GOTO 1090
1130 REM
1140 PT=I-1
1150 CLOSE #1
1160 RETURN
1500 REM
1510 REM AFFICHAGE
1520 REM
1530 IF PT=0 THEN 1570
1540 FOR I=1 TO PT
```

```
1550 PRINT D$(I)
1560 NEXT I
1570 PRINT:PRINT "POUR REVENIR AU MENU TAPEZ UNE TOUC
HE"
1580 IF INKEY$="" THEN 1580
1590 RETURN
2000 REM
2010 REM INSERTION
2020 REM
2030 PT=PT+1
2040 PRINT "ENREGISTREMENT N.";PT;"    "
2050 LINE INPUT D$(PT)
2060 INPUT "VOULEZ-VOUS CONTINUER (O/N)";R$
2070 IF R$="O" THEN 2030
2080 RETURN
2500 REM
2510 REM SUPPRESSION
2520 REM
2530 INPUT "NUMERO DE L'ENREGISTREMENT A SUPPRIMER";K
2540 IF K<0 OR K>PT THEN 2600
2545 PRINT "ENREGISTREMENT N.";K;" : ";D$(K)
2550 INPUT "SUPPRESSION CONFIRMEE (OUI/NON)";R$
2555 IF R$<>"OUI" THEN RETURN
2560 PT=PT-1
2565 FOR J=K TO PT
2570 D$(J)=D$(J+1)
2580 NEXT J
2590 RETURN
2600 PRINT "SUPPRESSION IMPOSSIBLE"
2610 PRINT:PRINT "TAPEZ UNE TOUCHE POUR CONTINUER "
2620 IF INKEY$="" THEN 2620
2630 RETURN
3000 REM
3010 REM ECRITURE SUR CASSETTE
3020 REM
3030 INPUT "LA   CASSETTE   EST-ELLE   EN  POSITION D
'ENREGISTREMENT (OUI/NON)";R$
3040 IF R$<>"OUI" OR PT=0 THEN RETURN
3050 INPUT "NOM DU FICHIER SORTIE :";FIO$
3070 OPEN "O",#1,FIO$
3080 FOR I=1 TO PT
3090 PRINT #1,D$(I)
3100 NEXT I
3110 CLOSE #1
3120 RETURN
```

EXERCICES POUR TO7

Variables

R - réponses au menu

K - numéro de l'enregistrement à supprimer

R$ - réponse

FII$ - nom du fichier lu

FIO$ - nom du fichier écrit

I,J - compteurs

D$(2ØØ) - tableau contenant la liste des enregistrements

PT - nombre d'enregistrements du fichier

Commentaires

1Ø-14Ø : Programme principal comprenant le menu d'aiguillage sur les différents sous-programmes.

1ØØØ-116Ø : Sous-programme de lecture du fichier à partir de la cassette.

1Ø7Ø : L'instruction OPEN "I",# 1,FII$ affecte au canal 1 le fichier FII$ enregistré sur cassette et précise ("I") que l'échange des données s'effectuera de la cassette vers le micro-ordinateur.

11ØØ : La fonction EOF(1)-End of File- est utilisée pour indiquer la fin du fichier. Le chiffre 1 précise le numéro du canal.

111Ø : # 1 dans l'instruction LINE INPUT indique que l'entrée se fait à partir du canal 1.

115Ø : L'instruction CLOSE #1 ferme le fichier.

15ØØ-159Ø : Sous-programme d'affichage du contenu du fichier.

2ØØØ-2Ø8Ø : Sous-programme d'insertion d'enregistrements.

25ØØ-263Ø : Sous-programme de suppression d'enregistrements.

2565-258Ø : Après suppression d'un enregistrement, tous les enregistrements suivants sont décalés d'un rang vers le haut.

3ØØØ-312Ø : Sous-programme d'enregistrement du fichier sur la cassette.

3Ø7Ø : L'instruction OPEN"O", 1,FIOS affecte au canal 1 le fichier FIO$ à enregistrer sur la cassette et précise ("O") que l'échange

des données s'effectuera du micro-ordinateur vers la cassette.

3090 : #1 dans l'instruction PRINT indique que l'envoi se fait vers le canal 1.

3110 : L'instruction CLOSE#1 ferme le fichier.

EXERCICE 5.8

Programme

```
10 CLS
15 CLEAR ,,2
20 SCREEN 7,0,1
25 DEFGR$(0)=0,0,4,2,255,2,4,0
30 DEFGR$(1)=0,0,32,64,255,64,32,0
35 DATA ROUGE,VERT,JAUNE,BLEU,MAGENTA,CYAN,WHITE
40 TITRE$="MASTERMIND"
45 B$="RVJBMCW"
50 NBCOL=7
60 NBPION=5
65 BOX(7,7)-(29,9)"*"
70 FOR I=1 TO 10
80 PSET (2*I+7,8) MID$(TITRE$,I,1),(I MOD7)+1,0
85 NEXT I
90 COLOR 7,0
100 LOCATE 5,15
110 PRINT "SI VOUS ETES PRET TAPEZ O   ";
120 REM TT SERT A INITIALISER RND
130 TT=TT+1:IF INKEY$<>"O" THEN 130
150 PRINT "O":PRINT
155 PRINT TAB(5)"COMBIEN VOULEZ-VOUS D'ESSAIS";:INPUT
  NBCOUP:PRINT:PRINT
160 IF NBCOUP=0 THEN 155
165 REM AFFICHAGE DES REGLES DU JEU
170 RESTORE
175 FOR I=1 TO NBCOL
180 READ C$
185 COLOR I
190 PRINT TAB(10)C$;TAB(20)"TAPEZ ";LEFT$(C$,1)
200 NEXT I
210 COLOR 7
220 PRINT TAB(6)"AUTRES TOUCHES  :  "+GR$(0)+"  "GR$(1)+
"  ";:COLOR 0,3:PRINT "ENTREE";:COLOR 7,0
225 PRINT:PRINT
230 PRINT TAB(8)"B SIGNIFIE BIEN PLACE"
240 PRINT TAB(8)"M SIGNIFIE  MAL PLACE"
250 PRINT:PRINT:
```

```
255 GOSUB 3000 'CHOIX DE LA CONFIGURATION A DECOUVRIR
260 PRINT "ESSAI"TAB(20)"REPONSE":PRINT
265 LOCATE 0,CSRLIN,0
270 FOR I=1 TO NBCOUP
280 PRINT I;
290 GOSUB 1000 'SAISIE DU COUP JOUE
300 GOSUB 2000 'REPONSE
310 PRINT:PRINT
320 IF NBB=5 THEN ATTRB 1,1:PRINT:PRINT:PRINT TAB(6)"
GAGNE":ATTRB 0,0:GOTO 400
330 NEXT I
340 PRINT:PRINT "VOUS AVEZ PERDU. POUR GAGNER IL FALL
AIT JOUER :"
350 PLAY "SODOSIDORE"
360 REM AFFICHAGE DE LA COMBINAISON A DECOUVRIR
370 FOR I=1 TO NBPION
380 PSET(3*I+15,CSRLIN)" ",INSTR(1,B$,R$(I)),0,0
390 NEXT I
400 COLOR 7,0
410 END
1000 REM SAISIE DU COUP JOUE
1005 NULI=CSRLIN
1010 FOR J=1 TO NBPION
1020 A$(J)="W"
1030 PSET(3*J,NULI)" ",7,0,0
1050 NEXT J
1060 J=1
1070 PSET(3*J,NULI)"?",INSTR(1,B$,A$(J)),0,0
1080 REM ENTREE DE CJ$ ET TEST POUR LE RANGER DANS A$
(J) OU DEPLACEMENT DU CURSEUR
1110 CJ$=INPUT$(1)
1120 L=INSTR(1,B$,CJ$)
1130 CASC=ASC(CJ$)
1140 IF CASC<>8 AND CASC<>9 AND CASC<>13THEN IF L=0 T
HEN 1110 ELSE A$(J)=CJ$
1160 PSET(3*J,NULI)" ",INSTR(1,B$,A$(J)),0,0
1180 IF CASC=13 THEN COLOR7,0:RETURN
1190 IF CASC=8 THEN J=J-2
1200 J=J+1
1210 IF J>5 THEN J=5
1220 IF J<1 THEN J=1
1230 GOTO 1070
2000 REM REPONSE AU COUP JOUE
2010 NBB=0:NBM=0 'INITIALISATION
2015 REM CALCUL DU NOMBRE DE PIONS BIEN PLACES
2020 FOR H=1 TO NBPION
2030 IF R$(H)=A$(H) THEN NBB=NBB+1:A$(H)="*"
2040 NEXT H
```

```
2050 FOR H=1 TO NBPION
2060 IF A$(H)="*" THEN 2110
2065 REM CALCUL DU NOMBRE DE PIONS MAL PLACES
2070 FOR K=1 TO NBPION
2090 IF A$(K)=R$(H) THEN NBM=NBM+1:A$(K)="/":GOTO 211
0
2100 NEXT K
2110 NEXT H
2112 REM AFFICHAGE DES RESULTATS
2115 LOCATE 20,CSRLIN
2120 FOR T=1 TO NBM
2130 PRINT "M  ";
2140 NEXT T
2150 FOR T=1 TO NBB
2160 PRINT "B  ";
2170 NEXT T
2180 RETURN
3000 REM CHOIX DE LA CONFIGURATION A DECOUVRIR
3005 TT=RND(-TT)
3010 FOR I=1 TO NBPION
3020 P=INT(7*RND)+1
3040 R$(I)=MID$(B$,P,1)
3050 NEXT I
3060 RETURN
```

Variables

TITRES	- chaîne de caractères
B$	- initiales des couleurs utilisées
NBCOL	- nombre de couleurs
NBPION	- nombre de pions
NBCOUP	- nombre d'essais
C$	- couleur
H,I,J,K,T	- compteurs
NBB	- nombre de pions bien placés
NBM	- nombre de pions mal placés
R$(5)	- tableau de la combinaison à découvrir
A$(5)	- essai proposé par le joueur
NULI	- numéro de ligne du curseur
CJ$	- caractère tapé par le joueur
L	- position du caractère tapé dans B$
CASC	- code ASCII du caractère tapé

Commentaires

25-3Ø	: Définitions des caractères "flèches".
45	: La lettre B étant utilisée pour la couleur bleue, on a choisi la lettre W (white) pour la couleur blanche .
65-85	: Affichage du titre
165-25Ø	: Affichage des règles du jeu .
17Ø-2ØØ	: Affichage de la correspondance touche/couleur .
22Ø	: Affichage des autres touches utilisables .
27Ø-33Ø	: Saisie et traitement des essais .
1ØØØ-123Ø	: Sous-programme de la saisie d'un essai .
1Ø1Ø-1Ø5Ø	: La combinaison initiale est composée de cinq pions blancs .
112Ø	: Si le caractère tapé CJ$ n'appartient pas à B$, L sera égal à Ø.
114Ø	: 8,9 et 13 sont les codes ASCII respectifs de "flèche-à-gauche", "flèche-à-droite" et "ENTREE".
116Ø	: Suppression du point d'interrogation dans la case validée.
118Ø	: La saisie de l'essai est terminée si le joueur tape sur la touche "ENTREE".
2ØØØ-218Ø	: Sous-programme d'analyse de la combinaison proposée .
3ØØØ-3Ø6Ø	: Sous-programme de choix de la configuration à découvrir .

EXERCICE 5.9

Programme

```
10 CLS
20 CLEAR ,,1:DEFINT A-Z
30 DEFGR$(0)=24,60,126,255,255,255,255,126
40 COLOR 1,3:ATTRB 1,1:LOCATE 7,4
50 PRINT "JEU DE NIM":COLOR 7,0:ATTRB 0,0:PRINT
60 INPUT "NOM DU PREMIER JOUEUR";N$(0)
70 INPUT "NOM DU DEUXIEME JOUEUR";N$(1)
80 SCREEN ,,1:CLS:LOCATE 0,0,0
```

```
120 REM
130 REM INITIALISATION DU JEU
140 REM
150 GOSUB 1500
160 T(1)=1:T(2)=3:T(3)=5:T(4)=7:SOM=16
170 REM
180 REM DEBUT DU JEU
190 REM
200 FOR K=0 TO 15
210 NJ$=N$(K MOD 2)
220 GOSUB 2500 'NJ$ JOUE
230 GOSUB 2000 'EFFACEMENT
240 IF SOM<=1 THEN 500
250 NEXT K
500 REM
510 REM NOM DU GAGNANT
515 IF SOM=0 THEN NJ$=N$((K+1) MOD 2)
520 COLOR 1,7 :ATTRB 1,1:LOCATE 10,5
530 PRINT NJ$+" A GAGNE"
540 ATTRB 0,0:COLOR 7,0
550 LOCATE 0,22:END
1500 REM
1505 REM AFFICHAGE INITIAL DES ALLUMETTES
1510 REM
1515 FOR J=1 TO 4
1520 FOR I=1 TO 9-2*J
1525 LINE (2*I,25-5*J)-(2*I,22-5*J) " ",0,7
1530 PSET (2*I,21-5*J) GR$(0),1,0
1540 NEXT I,J
1550 COLOR 7,0
1560 RETURN
2000 REM
2005 REM EFFACEMENT DES ALLUMETTES
2010 REM DE LA LIGNE LIG
2020 REM
2030 FOR I=T(LIG) TO T(LIG)-NBAL+1 STEP -1
2040 LINE (2*I,5*LIG)-(2*I,-4+5*LIG) " ",0,0
2050 NEXT I
2060 COLOR 7,0
2070 T(LIG)=T(LIG)-NBAL
2075 SOM=SOM-NBAL
2080 RETURN
2500 REM
2505 REM ACQUISITION D'UN COUP
2510 REM
2515 LOCATE 20,5:PRINT NJ$+" JOUE"
2520 LOCATE 20,6,1:PRINT "NUMERO DE LIGNE "+CHR$(24);
:INPUT LIG
```

```
2525 LOCATE 15,7:PRINT "NOMBRE D'ALLUMETTES"+CHR$(24)
;:INPUT NBAL
2530 IF NBAL>T(LIG) THEN LOCATE 20,8,0:PRINT "REJOUEZ
":PLAY "SIDO":GOTO 2520
2540 FOR I=5 TO 8
2550 LINE (15,I)-(38,I) " "
2560 NEXT I
2570 RETURN
```

Variables

N$(1)	- tableau des noms des deux joueurs
T(4)	- tableau des nombres d'allumettes par ligne
SOM	- nombre d'allumettes restantes
NJ$	- nom du joueur
NBAL	- nombre d'allumettes
I,J,K	- compteurs
LIG	- numéro de ligne

Commentaires

2Ø	: L'instruction DEFINT A-Z précise que toutes les variables utilisées seront entières.
3Ø	: Définition du dessin du bout de l'allumette.
2ØØ-25Ø	: Chaque joueur joue en alternance (K MOD 2 prend alternativement la valeur Ø et 1).
515	: Le perdant est le joueur qui prend la dernière allumette.

Achevé d'imprimer en avril 1984
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : avril 1984

N° d'impression : 403064
N° d'édition : 86595-141-1
ISBN : 2-86595-147-2